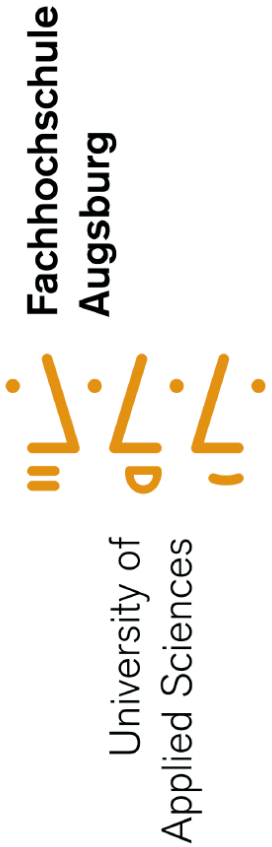
Einführung in die Standards IEEE 802.15.4 und ZigBee

Diplomarbeit

Michael Brenner

Studienrichtung

Elektrotechnik



Erstprüfer: Prof. Dr. Hubert Högl

Zweitprüfer: Prof. Dr. Martin Bayer

Einführung in die Standards IEEE 802.15.4 und ZigBee

Diplomarbeit

Michael Brenner

Studienrichtung

Elektrotechnik

Einführung in die Standards IEEE 802.15.4 und ZigBee anhand des Beispiels „Drahtloses Abstimmgerät“

Erstprüfer: Prof. Dr. Hubert Högl

Zweitprüfer: Prof. Dr. Martin Bayer

Beginn: 19.06.2007
Abgabe: 17.12.2007

Verfasser der Diplomarbeit:
Michael Brenner
Augsburger Straße 15
D-86 833 Ettringen
Telefon: 0179 / 357 03 04
E-Mail: mik@gmx.org
Mat.-Nr. 403001

Fachbereich Elektrotechnik
Telefon: +49 821 5586-350
Fax: +49 821 5586-360

Fachhochschule Augsburg
University of Applied Sciences
Baumgartnerstraße 16
D-86 161 Augsburg

Telefon: +49 821 5586-0
Fax: +49 821 5586-222
E-Mail: poststelle@fh-augsburg.de
http://www.fh-augsburg.de

Danksagung

Hier mit möchte ich mich bei Professor Högl bedanken. Er gab mir Gelegenheit über dieses interessante Thema zu schreiben. Weiteren Dank gilt ihm, weil er die schwer erhältliche Hardware der ZigBee-Module für mich organisiert hat und darüber hinaus mir größte Freiheit bei der Arbeit lies. Auch von zweifelsohne unschätzbarem Wert ist, dass die IEEE viele ihrer Standards kostenfrei zum Download verfügbar machen. Das Unternehmen Meshnetics sollte hier nicht unerwähnt bleiben. Sie stellt ihren OpenMAC unter einer Opensource-Lizenz zur Verfügung und ermöglicht vertiefte praktische Einblicke in die Umsetzung der Standards. Ein kleiner aber sicher bedeutenden Beitrag leistet Christian Richter aus dem www.mikrocontroller.net-Forum als er mich darauf hinwies, dass 2,4-GHz-Band Koexistenz-Probleme mit Bluetooth vorliegen. Allgemein sei den Professoren des Fachbereich Elektrotechnik gedankt der anregende Vorlesungen ein wichtiger Beitrag zu meinem Werdegang sind.

Lizenz

Diese Datei wurde unter den Bedingungen der „Creative Commons Namensnennung-Weitergabe unter gleichen Bedingungen Unported“-Lizenz (abgekürzt „cc-by-sa“) in der Version 3.0 veröffentlicht.

Inhaltsverzeichnis

1 Einführung.....5

2 Funkstrecke.....6

2.1 Störeinflüsse.....6
2.2 Modell der Pfadverluste.....8
2.3 Antennen.....9

3 Chips und Module.....11

3.1 Transceiver-Chips.....11
3.2 Module.....13
3.3 Software.....14
3.4 Firmware.....15
3.5 Software-Bibliotheken.....15
3.6 Eigenentwicklung.....16
3.7 ZigBee und Zertifizierung.....16
3.8 MAC-Adressen.....16

4 Schichtenmodell.....18

4.1 Protokollstapel.....18
4.2 Aufbau einer Schicht.....20
4.3 Konzept der Primitives.....21
4.4 Datenkapselung.....22

5 Der Laboraufbau.....23

5.1 Pinbelegung.....23
5.2 ZigBit.....24

6 Ansteuerung des Transceiver-ICs.....28

7 Der MAC-Layer.....35

7.1 Data-SAP des MAC-Layer.....35
7.2 Management-SAP des MAC-Layer.....38
7.3 MAC-PIB.....41

8 Beispiel für die Anwendung des MAC-Layers.....44

8.1 Entwicklungsumgebung.....44
8.2 Erstellen eines eigenen Projekts.....47
8.3 Starten des MAC-Layers.....47
8.4 Funktionsweise des Coordinator.....48
8.5 Associate.....50
8.6 Datenübertragung.....52
8.7 Funktion des Voters.....54

9 Ausblick.....57

10 Literatur.....58

10.1 Einzelnachweise.....58

11 Abkürzungsverzeichnis.....60

12 Index.....61

13 Daten-CD.....62

14 Erklärung.....63

1 Einführung

Die Computerwelt macht es vor, drahtlose Datenübertragung wird immer beliebter. In naher Zukunft sollen in Gebäuden, gewerbliche und privat, alle Sensoren, vom Feuermelder bis zum Lichtschalter, und alle Aktoren, von der Heizungssteuerung bis zum Türöffner, über Funk gesteuert werden können.

Ein Kandidat der antritt diese Aufgabe zu lösen ist der ZigBee-Standard. Doch selbst 3 Jahre nach Veröffentlichung sind nur wenige Anwendungen in Sicht. Gründe hierfür liegen in der hohen Komplexität, hinzu kommen Irrtümer, Inflationärer Gebrauch des Begriffs ZigBee und nicht genau geklärte Koexistenzprobleme mit anderen Sendern (WLAN, Bluetooth).

Unmittelbar mit ZigBee verbunden ist der IEEE-Standard 802.15.4. Auf diesen baut ZigBee auf. Viele wichtige Eigenschaften hängen von ihm ab. Dazu gehören stromsparendes Arbeiten, robuste Modulation, so dass hohe Reichweiten und gute Reaktionsgeschwindigkeit möglich sind.

Damit ein Überblick möglich wird, spannt diese Diplomarbeit einen weiten Bogen. Er reicht von den Fragen der Hochfrequenztechnik. Der anschließende Abschnitt geht auf mögliche Koexistenzprobleme mit WLAN ein und klärt die Maßnahmen mit denen der Standard Konkurrenz-Sender berücksichtigt. Es wird ein Marktüberblick für geeignete Hardware gegeben. Anhand des Schichtenmodells wird allgemein gefasst durchgenommen welche Aufgaben die einzelnen Ebenen haben.

Ein kurzes Beispiels zeigt welche Funktionen die Hardware bietet. Darauf folgend wird der MAC-Layer, die Trennstelle zwischen 802.15.4 und ZigBee, im Detail behandelt. Das anschließende Beispiel „Drahtloses Abstimmgerät“ verdeutlicht die korrekte Ansteuerung des MAC-Layers.

Auf der Ebene von ZigBee wird nicht derart ins Detail gegangen. Interessenten an einer zuverlässigen Funklösung dürften auf Ebene des MAC-Layers erst einmal befriedigt sein. Denn ZigBee wird, bis sich diese Technologie etabliert hat, für viele Anwendungen zu teuer (Software Lizenzen) und aufwendig sein.

2 Funkstrecke

Die nach IEEE 802.15.4 festgelegten Frequenzen befinden sich alle in den ISM-Bändern. Da die Frequenzen lizenzfrei sind muss auch mit anderen Sendern, wie WLAN gerechnet werden. Für diesen Fall führt der Standard IEEE 802.15.4-2003[1] auf mit den Ergebnissen, welche Störungen entstehen, allerdings basieren alle nur auf Simulation. (Zur kürze wird weiter nur auf das 2,4-GHz-Band eingegangen)

2.1 Störeinflüsse

Zu den Störern im 2,4-GHz-Band gehören:

- 802.11b (WLAN für drahtloses Video, Radio und Internet)
- 802.15.1 (Bluetooth für Freisprecheinrichtungen und Synchronisation von Organzier)
- 802.15.3 (sollte UWB werden, wurde eingestellt)
- Keinem Standard zugehörendes Funk-Video und Audio

Für Anwendungen mit hohen Ansprüchen an die Zuverlässigkeit empfiehlt es sich weniger stark belegt Frequenzbänder zu nutzen. 802.15.4 sieht hierfür das 868/915-MHz-Band vor, dessen Benutzung zusätzlich Reglementiert ist. Darum ist die Störung durch andere Sender selten[2].

WLAN

Eine WLAN-Transmitter verfügt über eine Sendeleistung von bis zu 14 dBm verteilt auf 20 MHz Bandbreite. Somit sendet dieser deutlich stärker als 802.15.4 (meistens 0 dBm), der verteilt diese jedoch auf ein schmäleres Frequenzband. Das Diagramm in Abbildung 4 zeigt die errechnete Fehlerrate, die bei einem Empfänger auftritt der -75 dBm Nutzsignalleistung erhält aber in dessen Umfeld sich ein WLAN-Sender befindet. Nach dem Path-Loss-Model (weiter unten mehr dazu) ergäbe sich für den 15.4-Sender ein Abstand von 25 Metern zum Empfänger. Selbst wenn der Störer auf einer benachbarten Frequenz arbeitet tritt eine Erhöhung der Übertragungsfehler auf. Bei Frequenzüberlappung sollte das störende WLAN die vierfache Distanz des 15.4-Senders aufweisen damit keine Störungen auftreten können. Als Gegenmaßnahme biete der Standard das Clear-Channel-Assessment (CCA, dt. Kanalfrei-Test), bei dem nur gesendet wird wenn kein anderer sendet. Hinzu kommt ein ACK-Paket mit dem eine erfolgreiche

Übertragung sofort bestätigt wird, ansonsten wird nochmals gesendet. Das Problem kann somit teilweise umgangen werden. Nach Möglichkeit sollte eine Frequenzüberlappung mit einem parallel genutzten WLAN vermieden werden. WLANs in der Nachbarschaft lassen sich in vielen Fällen hinnehmen. Problematisch wird es in Mehrparteienhäusern und gewerblichen Betrieben. Hier können zeitweise alle Kanäle belegt und ausgelastet. Die reale Reichweite sinkt.

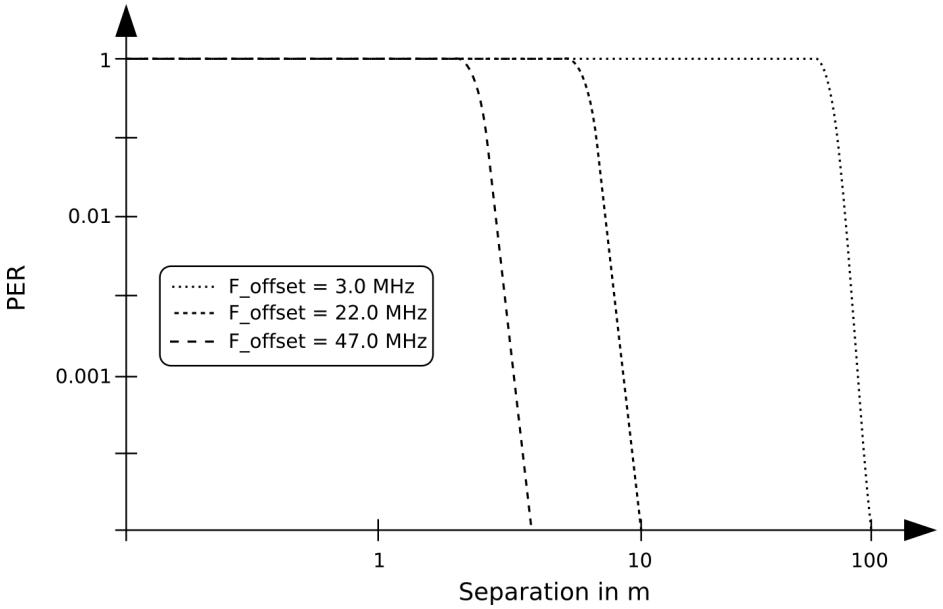


Abbildung 1: Package-Error-Rate bei WLAN mit gleichem Kanal und als Nachbarkanal (entspricht IEEE 802.15.4-2003 Fiugre E.3)

Bluetooth

Bluetooth macht nach Angaben des Standards nur geringe Probleme. Selbst unter ungünstigen Bedingungen tritt keine größere PER (Package-Error-Rate) als 10 % auf. Hier reichen die Maßnahmen zur Übertragungssicherung des Standards aus. Sollte es zu einem Problem kommen wird die Übertragung wiederholt. Allerdings gibt es seit Bluetooth V1.2 das adaptive Frequenzhopping (AFH). Es weicht WLAN aus und nutzt weniger Frequenzen. Aus diesem Grund kollidiert es dann häufiger mit den Übertragungsversuchen von 802.15.4. Ein aktives Bluetooth-Gerät direkt neben einem 802.15.4 Transceiver soll dessen Kommunikation komplett lahm legen.

Fazit

Die Nutzung des 2,4-GHz-Bandes birgt Risiken[3]. Konkurrenzstandards werben explizit damit, nicht das 2,4-GHz-Frequenzband zu nutzen[4]. Die Nutzung dieses Frequenzbandes wird in Zukunft noch stark zunehmen, da hier hohe Bandbreite und gute Ausbreitungseigenschaften zusammentreffen. Im Kontrast dazu sieht die ZigBee-Alliance geringe Bedenken[5].

2.2 Modell der Pfadverluste

Als Pfadverluste (engl. path loss) bezeichnet man die Dämpfung der Funkwellen im Raum aufgrund von Hindernissen und der Entfernung. Als Modell für die Pfadverluste gibt der Standard 802.15.4-2003[6] die Indoor-Dämpfung nach Gleichung 1 und Abbildung 2 vor. Diese gilt für zwei Bereiche mit Entfernungen über und unter 8 Meter.

$$L(\text{in dB}) = 40{,}2 + 20 \cdot lg(d) \;\; ; \;\; d \leq 8\text{m}$$

$$L(\text{in dB}) = 58{,}5 + 33 \cdot lg\left(\frac{d}{8\,m}\right) \;\; ; \;\; d > 8\,m$$

Gleichung 1: Pfadverluste für Indoor-Dämpfung

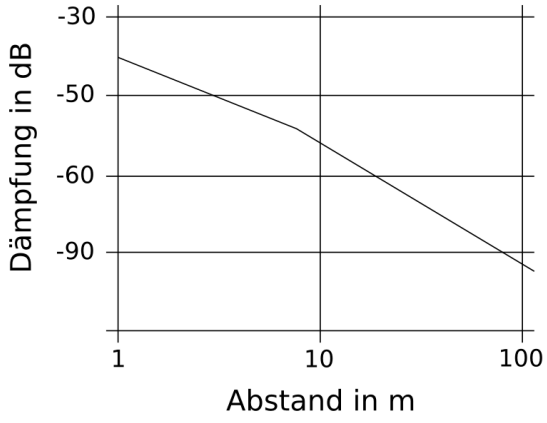


Abbildung 2: Grafische Darstellung der Pfadverluste

Die vorgegebenen Dämpfungen übersteigen deutlich die der Freiraumdämpfung. Berücksichtigt wurden Hindernisse wie Wände und Effekte der Mehrwegeausbreitung.

2.3 Antennen

Durch die geringe Anzahl an externen Bauteilen für den Empfänger bleibt der Anteil an Hochfrequenztechnik gering. Von Bedeutung ist vor allem die Auswahl der Antenne. Da der Empfang unabhängig von Ausrichtung des der Antenne sein sollte empfiehlt sich eine Antennencharakteristik mit Rundumempfang. Dank der hohen Frequenz erreicht man eine geringe Baugröße (Antennengröße liegt im Bereich von Lambda-Halbe, siehe Gleichung 2).

$$\lambda = \frac{c_0}{f} \cdot \frac{1}{\sqrt{\epsilon_r \cdot \mu_r}} = \frac{3{,}0 \cdot 10^8 \frac{m}{s}}{2{,}45\,GHz} = 122{,}5\,mm \;\Rightarrow\; \frac{\lambda}{2} = 61{,}25\,mm$$

Gleichung 2: Wellenlänge für 2,45 GHz

Koax-Anschluss

Möchte man die Entwicklung vereinfachen und auf das Design einer Antenne verzichten oder muss die Antenne an einem vom Transceiver entfernten Ort angebracht werden, führt man das Antennen-Signal einen Koax-Anschluss heraus. Da die meisten Transceiver einen symmetrischen Anschluss besitzen, muss dieser durch einen Balun für die asymmetrische Koax-Leitung angepasst werden[7].

Abbildung 3: Knick-Antenne für 2,45 GHz, annähernd Maßstabsgetreu

Als Antenne lassen sich die auch für WLAN üblichen Knick-Antennen verwenden (siehe auch Abbildung 3). Gleichwohl es äußerlich nicht ersichtlich ist, sind diese in der Regel

als Lambda-Halb-Dipol konstruiert. Damit sind die Knick-Antennen leicht zu handhaben.

Chipantennen

Eine kompakte und ohne all zu großen Aufwand realisierbare Antennenlösung sind die so genannte Chipantennen (siehe Abbildung 4). Im Datenblatt der Chipantennen ist deren Einbau und die Dimensionierung der Zuleitung genau definiert. Als Beispiel sei hier das Datenblatt zur „Compact Reach Extended“ des Herstellers Fractus genannt[8].

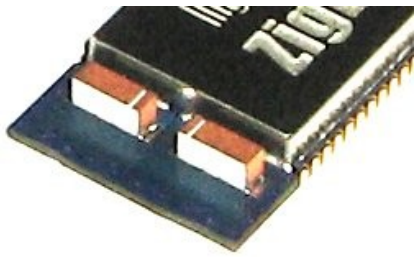


Abbildung 4: ZigBit-Modul,
Breite der Platine: 13 mm

Microstrip-Antennen

Vor allem Kostengründe sprechen für die Realisierung in Microstrip-Technik. Referenzen für symmetrische gespeiste Antenne liefern nahezu alle Hersteller der Transceiver-Chips. In den folgenden Datenblättern wird man fündig: Atmel Referenz-Kit[9], Meshnetics ZigBit[10] oder das Application-Note „Folded dipole antenna“ von Texas Instruments[11]. Wobei auf die richtige Impedanz der Antenne geachtet werden muss.

Asymmetrische Microstrip-Antennen sind zwar kleiner, das allerdings praktisch alle Transceiver einen symmetrischen Ausgang haben, hat seinen Grund. Sehr kompakte Geräte haben eine relativ kleine Massefläche, so dass ein „Gegengewicht“ zur Versorgung asymmetrisch gespeister Antennen nicht gegeben ist.

3 Chips und Module

Derzeit werden in zahlreichen Fachzeitschriften Module und Chips beworben, die ZigBee und IEEE 802.15.4 unterstützen. Einen Marktüberblick und welche Möglichkeiten die Bauteile wirklich bieten wird, soll in den nachfolgenden Abschnitten geklärt werden.

3.1 Transceiver-Chips

Die Transceiver-Chips bieten von sich aus weder 802.15.4 noch ZigBee. Das muss immer durch eine Mikrocontroller mit entsprechender Software erledigt werden. Jedoch liefern die Chips die Grundvoraussetzung damit die Standards eingehalten werden können. Einen Eindruck des Platzbedarf liefert Abbildung 5.

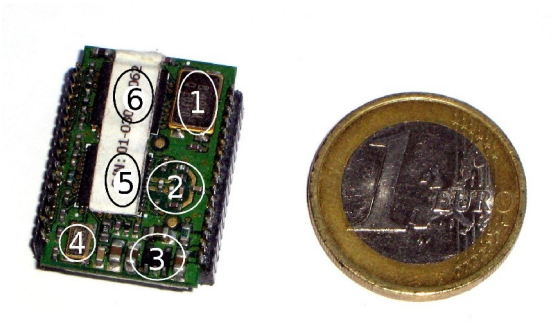


Abbildung 5:eazix ZigBee Platform

1: Ozsillator für Mikrocontroller
2: Balun auf 50 Ohm
3: Spannungsregler
4: Oszillator Transceiver
5: Transceiver
6: Mikrocontroller AVR ATmega32

Grundsätzlich führen die Chips eine korrekte Modulation und Demodulation der Signale aus. Des weiteren verfügen sie über einen mindestens 128 Byte großen FIFO-Puffer um einen kompletten Frame aufzunehmen. In diesem speichern sie die Daten zwischen bevor gesendet wird oder Puffern empfangenen Daten.

Weitere Grundfunktionen, die alle Chips bieten sind das automatische Prüfen der Checksumme (FCS, Frame-Check-Sequence) eingehender Frames bzw. deren

Generierung und das Anfügen beim Senden. Die automatische Generierung der ACK-Pakete hilft beim Einhalten des Timings. Da Verschlüsselungsalgorithmen in Hardware bedeuten schneller von statten gehen, liefern die Chips die vorgeschriebenen Methoden als integriertes Modul.

Von Texas Instruments kommt der Chip CC2420. Angesteuert wird dieser über eine SPI-Schnittstelle. Ein wichtiges Element ist der CCA-Pin, der anzeigt ob der Kanal gerade belegt ist oder nicht, etwa durch eine WLAN-Übertragung. Die schnelle Verfügbarkeit dieser Information ist für die Umsetzung des CSMA/CA-Algorithmus von Bedeutung. Die zweite Option ist der CC2430 der gleichzeitig einen C51-Prozessor enthält und somit eine SoC-Lösung darstellt.

Atmel bietet mit dem AT86RF230 einen über SPI angesteuerten Transceiver an. Mit der integrierten Funktionalität geht Atmel noch einen Schritt weiter. Den in IEEE 802.15.4 definierten CSMA/CA-Algorithmus übernimmt die Hardware, inklusive wiederholter Übertragungsversuche bei ausbleibendem ACK-Paket.

Den funktionsreichsten Baustein biete Microchip mit dem MRF24J40. Neben dem CSMA/CA-Algorithmus unterstützt die Hardware auch die so genannten Beacons und das Prinzip der Guaranteed-Timeslots (GTS). Diese Funktionen sind spezieller und werden nur bedingt benötigt. Für Fälle in denen sie benötigt werden reduzieren sich die Ansprüche an das Timing und die Software erheblich.

STMicroelectronics bietet in Zusammenarbeit mit Ember die Chips SN250 und SN260. Beides sind SoC-Lösungen für die Ember die notwendige Software entwickelt.

Freescale bietet die breiteste Palette an Chips. Dazu gehören Transceiver-Chips sowie SoC-Lösungen mit unterschiedlicher Speicherkapazität für maßgeschneiderte Skalierung.

Eine Ausnahme unter den Anbietern ist ZMD, die mehrere Chips für das 868/915-MHz-Band im Angebot haben. Neben der geringeren Konkurrenz durch andere Sender ergeben sich auch höhere Reichweiten bei dieser Frequenz, allerdings auch größeren Antennen.

3.2 Module

Neben den alleinigen Chips gibt es bereits fertige Module und sollen damit die Integration der Funkschnittstelle vereinfachen. Erster offensichtlicher Vorteil ist, das sich der Aufwand für die Entwicklung der Hochfrequenzkomponenten, wie Antenne oder EMV gerechter Schaltungsentwurf, stark verringert.

Zur Nutzung der reinen Hardware erfordern ZigBee und IEEE 802.15.4 einen beträchtlichen Softwareaufwand. Daher stellen Modul-Hersteller in der Regel passende SDKs und Beispielcode zur Verfügung, die den Einstieg erleichtern.

Viele Module gibt es in der bewährten Bauweise mit Stiftleisten (Beispiel Xbee von Maxstream). Als kostengünstigere und zudem kompaktere Lösung existieren Module die, einem SMD-IC ähnlich, direkt auf die Platine gelötet werden (Abbildung 7).

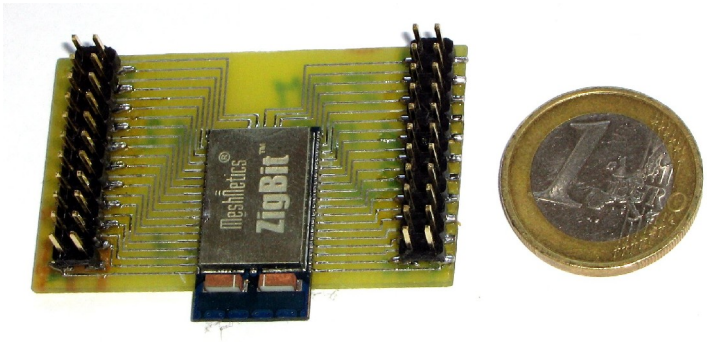


Abbildung 6: ZigBit-Modul von Meshnetics zum direkt Auflöten (hier mit Adapter-Platine)

Tabelle 1 liefert einen Überblick bedeutender Modul-Anbieter. Ein Modul besteht aus einem Mikrocontroller, dem Transceiver-IC und Antenne. Zusätzliche Sensoren, die ein Modul auf ein Anwendung spezialisieren, sind nicht enthalten.

Hersteller	Prozessor	Transceiver	ZigBee-Zertifiziert
eazix[12]	ATmega128L	CC2420	
Ember[13]	SoC, Atmel AVR	EM250, EM260, EM2420	x
Jennic[14]	SoC	32-Bit RISC	x
MaxStream[15]	Freescale M9S8GT60	MC13193	x
Meshnetics[16]	ATmega1281L	AT86RF230	x
Radiopulse[17]	SoC (C51)	MG2400	x
Renesas[18]	Renesas M16C	CC2420, ZMD44102	x
Sentec[19]	SoC, HC08	MC13192, MC13213	(Freescale Stack)
Telegesis[20]	SoC	EM250	(Ember Stack)

Tabelle 1: Marktübersicht

Module und Evaluation-Kits sind bei kleiner Stückzahl günstiger und sehr gut geeignet Erfahrungen zu sammeln.

3.3 Software

Beim Einsatz der Module kann der Entwickler wählen, bis zu welcher Stufe er die Standards umsetzt (Abbildung 7). Bei einfachen Lösungen ist es möglich, ohne jeden Standard die Hardware zu nutzen und lediglich etwa den CSMA/CA-Algorithmus aus 802.15.4 zu übernehmen. Wenn häufiger auf Funklösungen zurückgegriffen wird, lohnt sich auch die Umsetzung des Standards 802.15.4, die eine vereinheitlichte Schnittstelle zur Übertragung von Daten darstellt und die Bildung kleiner Netzwerke vereinfacht. Falls es um Fragen der Interoperabilität und Herstellerunabhängigkeit geht kommt nur noch die Umsetzung des ZigBee-Standards in Frage.

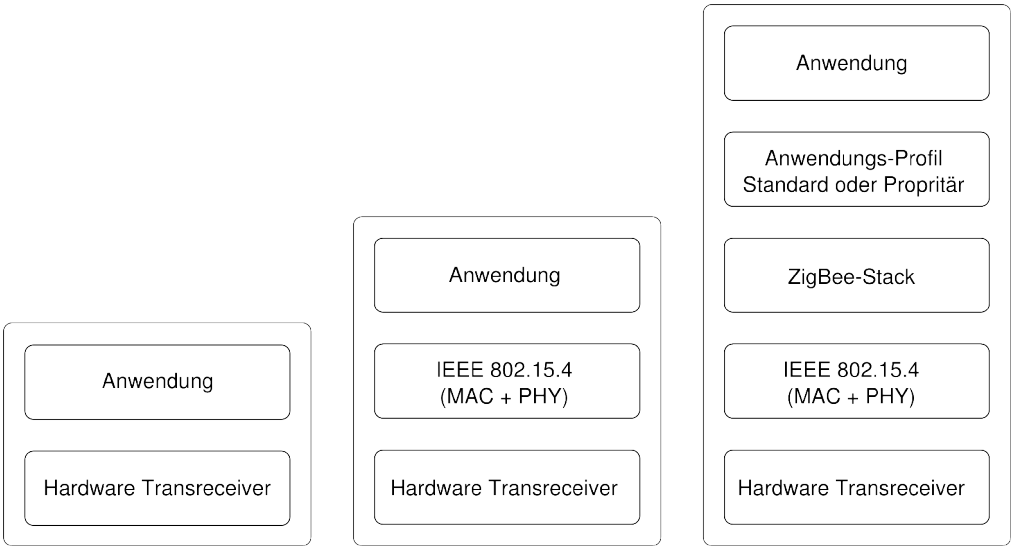


Abbildung 7: Nutzungsstufen

3.4 Firmware

Den minimalsten Aufwand zur Umsetzung eines ZigBee-Gerätes stellen vorprogrammierte Module. Die Ansteuerung erfolgt über eine serielle Schnittstelle. Dies biete zum Beispiel Xbee von Maxstream oder SerialNet von Meshnetics. Die Ansteuerung erfolgt über die am AT-Befehlssatz angelehnten Steuerzeichen, wie auch von Modems bekannt. Wesentliche Kenntnisse über ZigBee sind trotz allem erforderlich.

3.5 Software-Bibliotheken

Ein mit fester Firmware programmiertes Gerät benötigt noch einen weiteren externen Mikrocontroller. Zur Reduzierung der Bauteile kann auch in den für den ZigBee-Stack zuständigen Prozessor zusätzlich Software eingefügt werden. Hierfür stehen teilweise von den Herstellern Bibliotheken zur Verfügung. Zu beachten ist, Modul-Anbieter verlangen für ihren Stack neben dem Erwerb der Hardware zusätzlich Lizenzgebühren.

Ausnahme bilden der OpenMAC-Stack unter der der Opensouce-Lizenz CDDL und der MAC-Stack von Atmel. MicroChip bietet einen quelloffenen Stack, dessen Code hingegen wirkt sehr unstrukturiert, der nicht überzeugt. Texas Instruments und Freescale bieten einen lizenzfreien Stack, mit der Einschränkung, dass der Quellcode nicht offen vorliegt.

3.6 Eigenentwicklung

Aufgrund der hohen Komplexität ist die Eigenentwicklung eines ZigBee-Stacks nicht zu empfehlen. Selbst der MAC- und PHY-Layer nach Standard IEEE 802.15.4 benötigt schon mehr als 30 kB Programmcode bei nicht unerheblicher Komplexität. Allerdings hindert auch keiner einen daran die Hardware nach eigenen Regeln zu verwenden, solange das nicht in Konflikt mit dem Standard gerät.

Für eine zuverlässige Datenübertragung lassen sich einige im IEEE-Standard aufgeführte Techniken abwandeln. Hierzu gehören insbesondere der CSMA/CA-Algorithmus mit ACK-Paketen, das Frameformat und das Prinzip der Indirect-Transmission, welches für stromsparende Geräte praktisch ist.

3.7 ZigBee und Zertifizierung

Der ZigBee-Standard ist offen und die darin beschriebenen Funktionen unterliegen keiner Lizenz. ZigBee ist ein eingetragenes Markenzeichen, genauso auch das ZigBee-Logo. Durch ein einheitliches Zertifizierungs-Verfahren kann jeder Hersteller das Recht erwerben dieses Markenzeichen für sein Produkt zu verwenden. Die Zertifizierung stellt sicher, dass Geräte verschiedener Hersteller in einem Netzwerk coexistieren können[21].

Bei der Zertifizierung wird zwischen „ZigBee Compilant Platform“ und „ZigBee Certified“ unterschieden. „ZigBee Compilant Platform“ kennzeichnet eine Hardware in Verbindung mit eine ZigBee-Stack die es ermöglicht ZigBee-Produkte zu entwickeln. „ZigBee Certified“ bedeutet ein fertiges Produkt kann in einem ZigBee-Netzwerk teilnehmen. Weitere Kompatibilität mit Standardanwendungen, den sog. Profiles (wie Lichtschalter)[22] , ist nicht erforderlich.

Zur Zertifizierung befugt ist einmal die ZigBee-Alliance selbst. In Deutschland hat zu dem der TÜV die Berechtigung eine Zertifizierung durchzuführen. Weitere Voraussetzung ist die kostenpflichtige Mitgliedschaft bei der ZigBee-Alliance.

3.8 MAC-Adressen

Jeder standardkonforme Transceiver benötigt eine IEEE-64-Bit-Adresse, welche ähnliche Bedeutung wie die MAC-Adresse im Ethernet hat. Diese, EUI-64 (Extended-Unique-

Identifier) genannten, Adressen können bei der IEEE nur in Paketen erworben werden. Dabei kann eine OUI (Organisation-Unit-Identifier, 2^24 Adressen) gekauft werden oder auch nur ein IAB (Individual-Address-Block, 2^12 Adressen)[23]. Der Weiterverkauf einzelner Adressen oder die Übertragung an ein anderes Unternehmen ist nur sehr eingeschränkt möglich[24].

4 Schichtenmodell

ZigBee und IEEE 802.15.4 unterteilen die Aufgaben zum Informationsaustauschs in Schichten, welche an das ISO/OSI-Referenzmodell angelehnt sind. Der Modulare Aufbau erhöht die Flexibilität des Übertragungsprotokolls. Soll eine andere Hardware, z. B. Transceiver-Chip, als bisher eingesetzt werden sind nur Änderungen im PHY- und MAC-Layer notwendig. Die restliche Applikation kann unverändert übernommen werden.

4.1 Protokollstapel

Modular heißt dabei nicht das jedes Teil einzeln unabhängig von dem anderen Sinnvoll nutzbar ist. Grundsätzlich unterteilt sich der Protokollstack in zwei Einheiten, dem ZigBee-Standard und dem Standard IEEE 802.15.4 (Abbildung 8). Auch MiWi und andere Protokolle bauen auf dem IEEE-Standard auf.

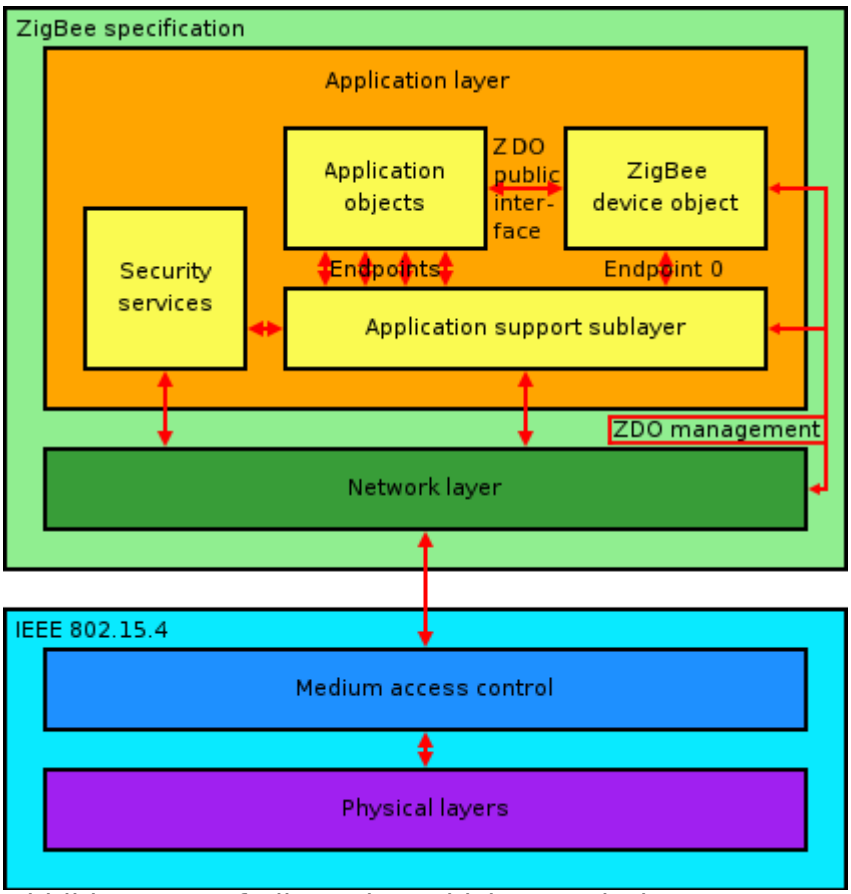


Abbildung 8: Aufteilung der Schichten zwischen IEEE 802.15.4 und ZigBee (Bild aus Wikiepdia, Public Domain)

Den anwendungsorientierten Teil erfüllen bei ZigBee das ZigBee-Device-Object (ZDO), der Application-Support-Sublayer (APS) und den Application-Objects.

Das ZDO sorgt für die Integration des Endgerätes in ein ZigBee-Netzwerk. Es legt die Rolle des Gerätes im Netzwerk fest, ob Coordinator, Router oder Enddevice. Es verbreitet die Gerätefunktionen im Netzwerk und initiiert das Binding, also der Verbindung der Gerätefunktionen mit anderen Geräten, z. B. Lichtschalter mit Lampe.

Application-Objects stellen die eigentliche Anwendung dar. Ihre Identität geben sie über ihren Device-Descriptor bekannt. Die Kommunikation erfolgt mit anderen Geräten anhand definierter Profiles.

Der APS verwaltet die Binding-Tabelle und sorgt somit für die Zustellung der verschickten Daten an das richtige Application-Object.

Zur Funktionalität des Network-Layers (NWK) gehört es, einen Router für den Eintritt in das Netzwerk auszuwählen und bei diesem sich in das Netzwerk zu registrieren. Auch ist der Layer verantwortlich für das Routing eingehender Datenpakete und Ermittlung aller direkter Nachbarn für Routingzwecke. Im Falle des Coordinators besteht die Aufgabe des NWK-Layers das Netzwerk zu starten.

Auf der Ebene des MAC-Layers wird die Zugriffsmethode auf den Kanal bestimmt. IEEE 802.15.4 greift hier auf mehrere Techniken zurück. Der CSMA/CA-Algorithmus soll möglichst Kollisionen vermeiden. Zum CSMA/CA-Algorithmus gehört optional ein Acknowledgement-Paket. Die Absicherung der Übertragung erfolgt mittels Prüfsumme und Verschlüsselung. Da mehrere Frequenzkanäle zur Auswahl stehen muss eine Funktion zum Scannen der Frequenzen verfügbar sein. In der Ausführung als Full-Function-Device (FFD) fungiert der MAC-Layer auch als PAN-Cooridnator und bietet Reduced-Function-Devices (RFD) und untergeordneten FFDs die Möglichkeit sich bei ihm zu registrieren. Im Gegensatzt zum NWK-Layer findet hier ausschließlich direkte (Point-to-Point) Kommunikation statt. Der Begriff PAN-Coordinator ist nicht mit dem Zigbee-Coordinator zu verwechseln. Sowohl ZigBee-Router als auch ZigBee-Coodinator agieren als PAN-Cooridnator. Bei bedarf sendet der MAC-Layer in regelmäßigen Zeitabständen Beacons (dt. Signalfeuer) aus, auf die sich untergeordnete Geräte synchronisieren.

Der Physical-Layer (PHY) gehört zur Bitübertragungsschicht. Das heißt hier werden die Daten von einem Gerät zum nächsten übertragen. Bei einer Funkschnittstelle sind aber noch weitere Aufgaben auszuführen. Es muss die Trägerfrequenz eingestellt werden auf der gesendet wird. Für dem CSMA/CA-Algorithmus muss das Clear-Channel-Asessment (CCA, = Kanalfrei Test) durchgeführt werden. Damit ein stromsparender Betrieb möglich ist kann der Sender ausgeschaltet werden.

4.2 Aufbau einer Schicht

Die Schichten des Protokollstacks sind gegeneinander abgegrenzte Einheiten. Der innere Aufbau folgt einem einheitlichen Schema. Der Austausch zwischen den Schichten erfolgt mit Schnittstellen, die man als Service-Access-Point (SAP) bezeichnet (Abbildung 9).

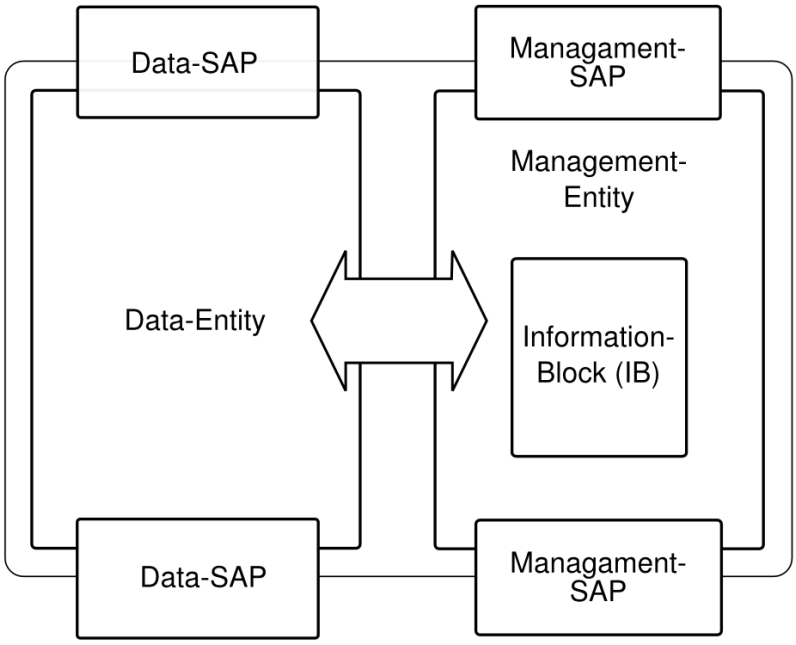


Abbildung 9: Innere Struktur einer Schicht

Bei den SAPs unterscheidet man zwischen der Datenschnittstelle und der Managementschnittstelle. Die Datenschnittstelle vermittelt Daten an die gleiche Schicht eines entfernten Gerätes. Als Ausnahme bietet der MAC-Layer dem NWK-Layer die Möglickeit Daten aus der Warteschlange zu entfernen (purgen) falls ein Teilnehmer aus dem Netzwerk aussteigen möchte. Auf der Managementschnittstelle stehen

weiterführende Dienste zur Verfügung. Sie liefern der darüberliegenden Schicht wichtige Informationen, die für deren Betrieb benötigt werden und die darüberliegende Schicht kann wichtige Parameter der unter ihr liegenden Schicht verändern, damit sie das gewünschte Verhalten aufweist. Zum Beispiel weißt der MAC-Layer an zwischen Sende- und Empfangsbetrieb umzuschalten und der PHY-Layer ist für die Ausführung zuständig.

Alle für den Betrieb einer Schicht notwendigen Parameter speichert der Information-Block (IB).

Neben der Teilung der Schnittstellen erfolgt auch eine funktionelle Teilung des Layer in eine Data-Entity und eine Management-Entity. Die Data-Entity sorgt für das Senden und Empfangen der Daten, dabei trennt es die Daten die eine anderes Geräte an den aktuellen Layer geschickt hat und gibt diese an das Management-Entity weiter. Das Management-Entity übernimmt die Verwaltungsaufgaben des Layers. Soll mit einem entsprechenden Layer auf einem entfernten Gerät kommuniziert werden gehen diese zum Data-Entity.

4.3 Konzept der Primitives

Der Standard definiert die Schnittstellen zwischen den Schichten nicht als Funktionsaufruf (API), sonder greift auf das abstraktere Konzept der Primitive-Passing-Technik zurück. Hierbei wird lediglich definiert welche Daten ausgetauscht werden, jedoch nicht auf welche Weise.

Die Primitives unterteilt man in vier Gruppen: Request, Indication, Response und Confirm. Abbildung 10 deckt den Zusammenhang zwischen den Primitives auf. Ein Request wird durch die darüberliegende Schicht gesendet. Nachdem ein Request abgearbeitet wurde schickt in der Regel die bearbeitende Schicht eine Confirm-Nachricht. Ein Indication wird auf Initiative der unteren Schicht ausgelöst. Die darüberliegende Schicht hat in der Regel darauf folgend mit einer Response-Nachricht zu reagieren. Auslöser für Indication-Nachrichten sind gewöhnlich eingehende Frames von anderen Geräten, die diesen als Reaktion auf einen Request verschicken.

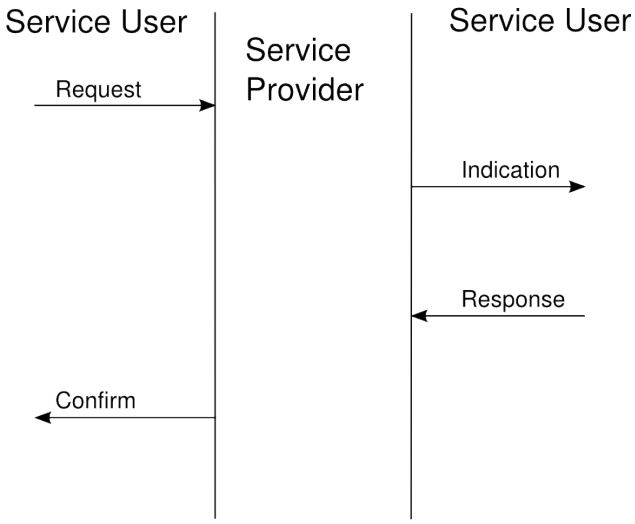


Abbildung 10: Konzept der Primitives

4.4 Datenkapselung

Beim weiterreichen der Daten von einer Schicht zur anderen fügt jede Schicht dem Datenpaket Verwaltungsinformationen hinzu (Abbildung 11). Daten die einer Schicht durch die Data-SAP übergeben werden bezeichnet man als Service-Data-Unit (SDU) im Fall des MAC-Layer MSDU. Daten denen Verwaltungsinformationen hinzugefügt werden (Header und Footer) bezeichnet man als Protocol-Data-Unit (PDU). Ein SDU kann auf manchen Schichten in mehrere PDUs aufgeteilt werden.

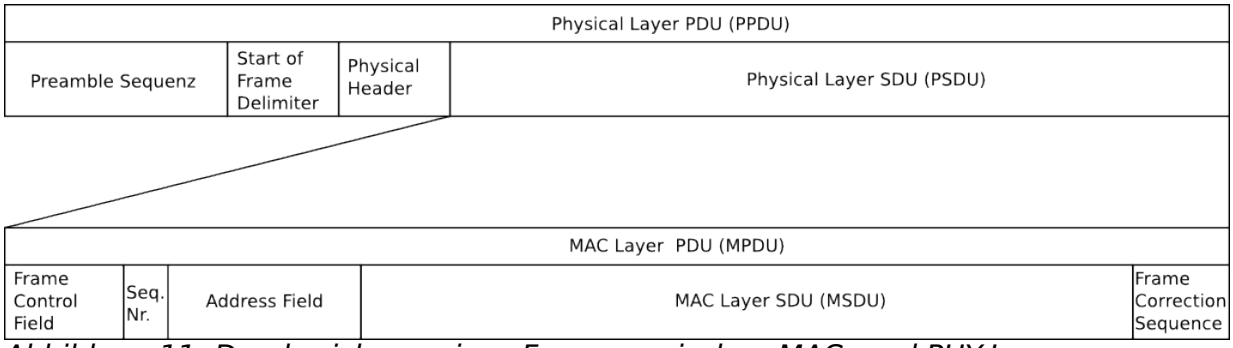


Abbildung 11: Durchreichung eines Frames zwischen MAC- und PHY-Layer

Das Schichtenmodell grenzt die die Aufgaben der einzelnen Ebenen klar ab. Aufgrund abstrakter Definitionen, wie Primitives, kann die Implementierung der Standards in sehr unterschiedlicher Weise, von Software- bis Hardwarelastig, erfolgen. Trotz allem ist die Zusammenarbeit der unterschiedlichen Geräte sichergestellt.

5 Der Laboraufbau

Als Grundlage des Projekts Drahtloses Abstimmgerät dient das Modul „ZigBit“ der Firma Meshnetics. Es enthält einen Mikrocontroller, einen Transceiver-Chip, einen Quarz und eine Antenne. Für den Betrieb sind noch externe Komponenten erforderlich. Die Abbildung 12 zeigt den Laboraufbau mit Lochraster.

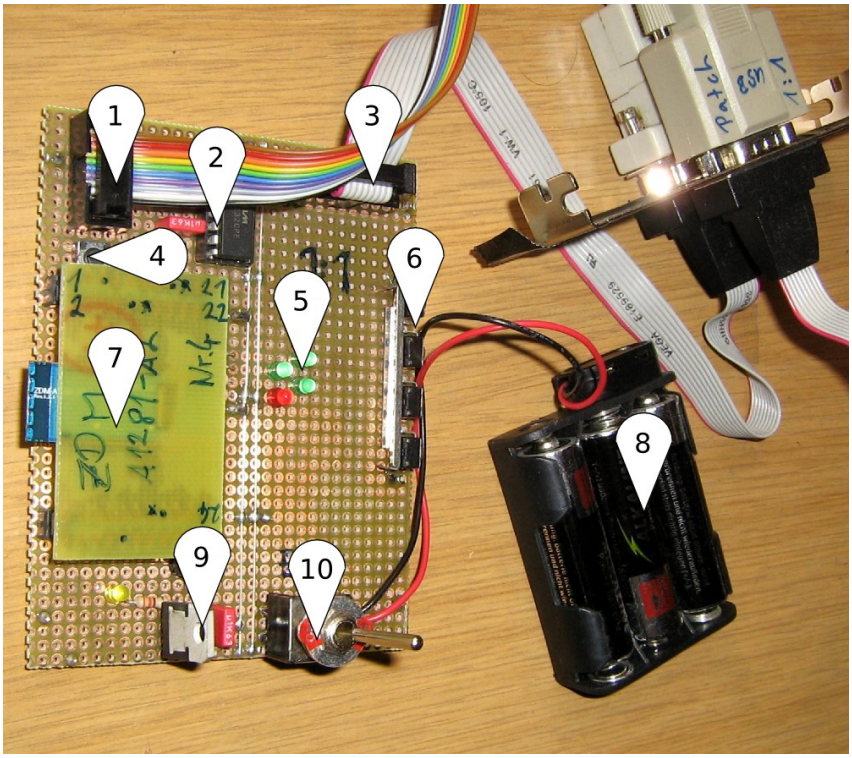


Abbildung 12: Laboraufbau
1 ISP-Anschluss für Programmer
2 MAX3232 Pegelanpassung zw. 3,3 V und RS232-Pegel
3 Übergang zu SUB-D9 Buchse
4 Reset-Taste für Mikrocontroller
5 Status-Leds
6 Taster
7 ZigBit-Modul auf Adapterplatine
8 Batterien zur Energieversorgung
9 Lowdrop-Linear-Regler LF33 mit Kondensatoren
10 Ein-/Aus-Schalter

5.1 Pinbelegung

Die Tabelle 2 gibt an mit welchen Pins die externen Komponenten verbunden sind. Nach Möglichkeit wurde darauf geachtet den Laboraufbau einfach zu halten und gleichzeitig

„wertvolle“ Anschlüsse zu schonen (wie das JTAG-Interface).

VerwendungsZweck	Bezeichnung Meshnetics	Bezeichnung AVR
LEDs:		
Rote LED	GPIO6	PD6
Erste grüne LED	GPIO7	PD7
Zweite grüne LED	GPIO0	PB5
Dritte grüne LED	GPIO1	PB6
Taster:		
Taster 1 (links)	USART0_EXT_CLK	PE2
Taster 2	GPIO8	PE3
Taster 3	IRQ_6	PE6
Taster 4 (rechts)	IRQ_7	PE7
ISP:		
PDI/MOSI	USART0_RXD	PE1
PD0/MISO	USART0_TXD	PE2
CLK	SPI_CLK	PB1
Reset	Reset	Reset
Reset:		
Reset-Taste	Reset	Reset
UART:		
RX	UART_RXD	PD2
TX	UART_TXD	PD3
RTS	UART_RTS	PD4
CTS	UART_CTS	PD5

Tabelle 2: Pinbelegung für externe Komponenten

5.2 ZigBit

Der im ZigBit-Modul verbaute Mikrocontroller ist der Atmel ATmega1281V und verfügt über 128 kByte Flash, 8 kByte RAM und 4 kByte Eeprom. Hinzu kommen zahlreiche Schnittstellen, wie I²C, Timer, A/D-Wandler und sparsame Low-Power-Modes[25]. Die Taktquelle für den Prozessor können entweder ein interner RC-Oszillator oder der vom Rf-Chip angezapfte Taktgenerator sein.

Programmiert man mit den Fuse-Bits den Rf-Chip als Taktquelle sind energiesparende Sleep-Modes bei denen der Taktgenerator abgeschaltet wird nicht mehr Nutzbar. Es besteht dann keine Möglichkeit zum Aufwachen aus dem Sleep-Mode. Der RC-Oszillator

ist immer noch genau genug für den Betrieb einer UART bis 38.400 Bits/s.

Hinweis: Nicht alle Pins des Mikrocontrollers sind herausgeführt. Insbesondere fehlt der Port A. Damit ist eine HV-Programmierung nicht durchführbar. Eine Programmierung ist nur über ISP-Schnittstelle oder JTAG möglich. Bei falsch gesetzten Fuses ist eine Rettung des Controllers nicht mehr möglich.

Als Transceiver-Chip kommt der AT86RF230 zum Einsatz. Der Chip ist für das 2,4 GHz-Band konzipiert. Die Sendeleistung beträgt 3 dBm (2 mW) und die Eingangsempfindlichkeit beträgt -101 dBm. Der Antennengewinn für die Integrierte Antenne beträgt 0 dBi. Zum Betrieb ist ein Schwingquarz mit 16 MHz im Modul integriert.

Die interne Verdrahtung des Transceivers mit dem Mikrocontroller zeigt Tabelle 3.

Pinbelegung intern	
AT86RF230	ATmega1281V
SS	PB0
SCLK	PB1
MOSI	PB2
MISO	PB3
RESET	PA7
CLKM	XTAL_IN
SLP_TR	PB4
IRQ	PE5

Tabelle 3: Interne Beschaltung

Pinbelegung des ZigBit-Modul

Abbildung 13 zeigt die Pinbelegung des ZigBit-Moduls. Angegeben ist jeweils die Bezeichnung, wie es die Software von Meshnetics verwendet und dazu die im Datenblatt des Mikrocontrollers benutzte Bezeichnung[26].

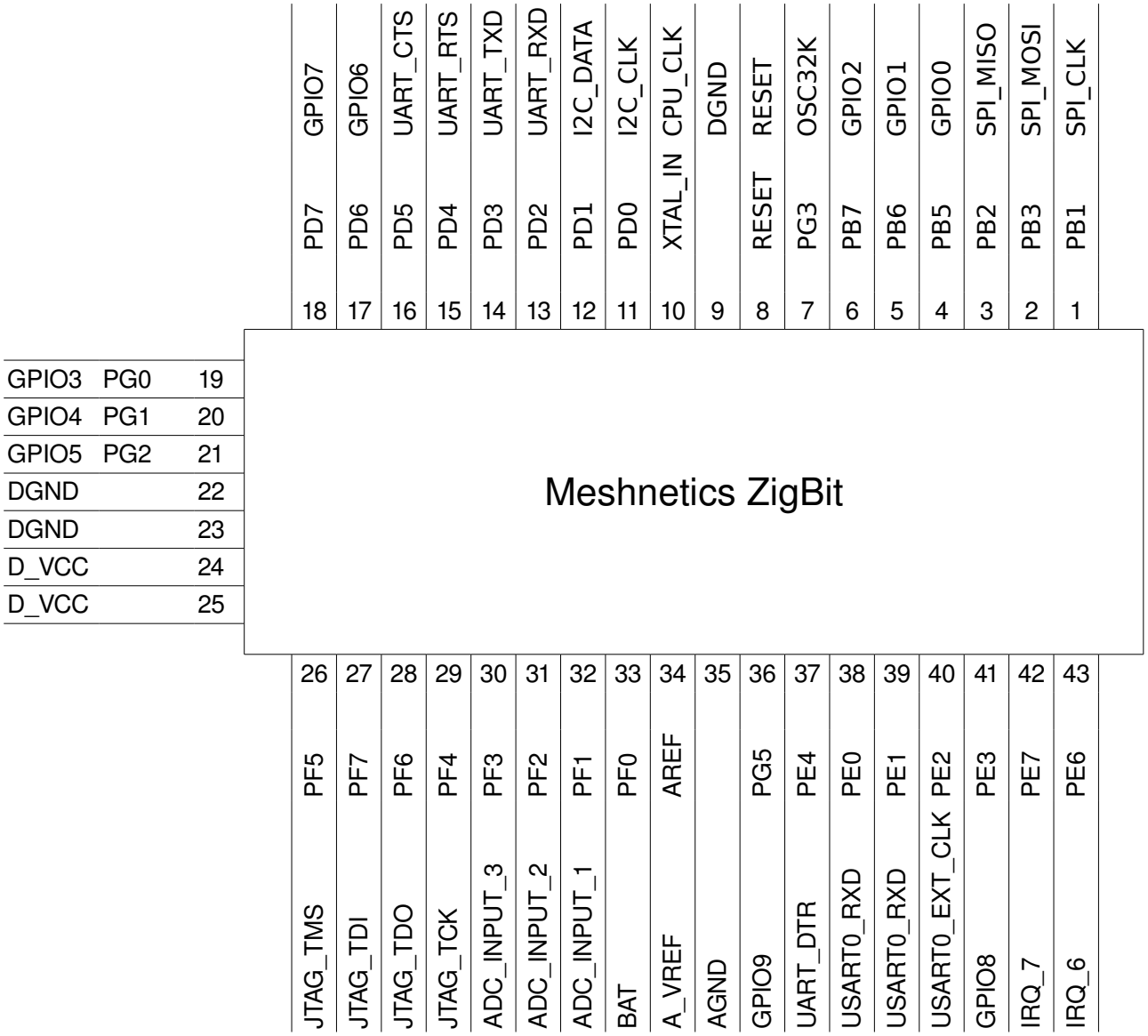


Abbildung 13: Pinbelegung des ZigBit-Moduls (Im Datenblatt sind UART_TXD und UART_RXD vertauscht)

Stromverbrauch

Das ZigBit-Modul ist für den Batteriebetrieb geeignet. Laut Datenblatt von Atmel benötigt der Transceiver 0,1 µA und der Controller ca. 6 µA im Power-Down, also mit deaktiviertem Taktgenerator. Im aktiven Zustand benötigt das Modul laut Meshnetics Datenblatt 19 mA während des Sendens, 17 mA bei Empfangsbereitschaft und 14 mA wenn nur der Controller aktiviert ist. Das Aufwecken des Transceiver erfolgt in weniger als 1 ms, damit kann ohne weiteres häufiger in den Schlafmodus geschaltet werden.

Angemerkt sein, dass Stromsparmaßnahmen bei ZigBee nur bedingt eingesetzt werden können. Nach Angabe von Maxstream, haben deren Modul 1000 Stunden (= 42 Tage) Batterielaufzeit. Auf die Anwendung zugeschnittene Lösungen können die Batterielaufzeit deutlich erhöhen.

Adapter

Für den Laboraufbau wurde eine Adapterplatine für das ZigBit-Modul angefertigt. Abbildung 14 zeigt das Layout von der Lötseite. Die äußeren Lötpads sind für die Montage von 2,54mm-SMD-Stiftleisten geeignet. Pin 1 liegt rechts unten außen, Pin 2 rechts unten innen usw.

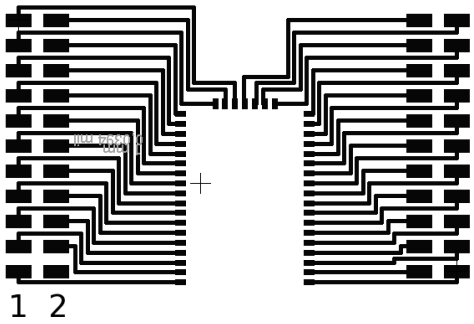


Abbildung 14: Adapter für ZigBit-Modul auf SMD-Stiftleiste mit 2,54mm-Raster

Für die Trägerplatine sind einige Vorschriften zu beachten. Auf der obersten Lage also unterhalb des Moduls darf keine Leiterbahn oder Durchkontaktierung liegen. Als Leiterplattenmaterial sollte FR4 mit der Stärke 1,6 mm verwendet werden. Die Träger-Leiterplatte sollte mit der Modul-Platine abschließen. Der Bereich der Antenne, also zwischen Platinenrand und den ersten Kontakten des Moduls, sollte auch links und rechts des Moduls von Bauteilen frei gehalten werden und keine Kupferkaschierung aufweisen. Details siehe im Datenblatt zum ZigBit-Modul[27].

Der Laboraufbau benötigt nur wenige externe Bauteile. Die Integration des Moduls ist anspruchslos. Damit liegt die Fokus für Anwendungen mit Modulen deutlich stärker auf der Softwareseite.

6 Ansteuerung des Transceiver-ICs

Transceiver-ICs liefern die Grundvoraussetzung um den Standard IEEE 802.15.4 zu erfüllen. Welche Funktionen bereits auf Ebene der Hardware ausgeführt werden liefert das kurze Beispiel einer 2-Kanal-Fernbedienung, realisiert mit dem Laboraufbau.

Die Funktionsweise des Beispiels lässt sich in wenigen Worten beschreiben. Der Sender überträgt alle 200 ms einen MAC-Frame der als Payload ein Byte beinhaltet, Bit 0 enthält den Zustand von Taster 1, Bit 1 den Zustand von Taster 2. Der Sender erkennt das kein Empfänger vorhanden ist, wenn Acknowledge-Pakete ausbleiben. Der Empfänger erkennt das der Sender fehlt, wenn 500 ms kein Paket mehr Empfangen wurde. Diesen Fall signalisieren beide durch aufleuchten der roten Led am Laboraufbau.

Vor der konkreten Betrachtung erfolgt ein allgemeiner Überblick. Abbildung 15 zeigt das vereinfachte Blockschaltbild des AT86RF230. Der Chip bietet eine volle SPI-to-Antenna-Lösung. Über die SPI-Schnittstelle können Register in der Control-Logic manipuliert sowie der Framebuffer gelesen und geschrieben werden. Die Modulation und Demodulation erfolgt in digitaler Signalverarbeitung. Im Analog-Teil erfolgt die Hoch- bzw. Tiefsetzung von der Sendefrequenz zur Zwischenfrequenz. Das Modulationsprinzip ist O-QPSK und Bandspreizung mit DSSS[28]. O-QPSK ist resistent gegen nichtlineare Verzerrungen und sorgt für effiziente Kanalausnutzung. DSSS erzeugt Redundanz und macht das Signal unempfindlich gegen Störungen[29]. Die Zwischenfrequenz (IF) liegt bei 2 MHz.

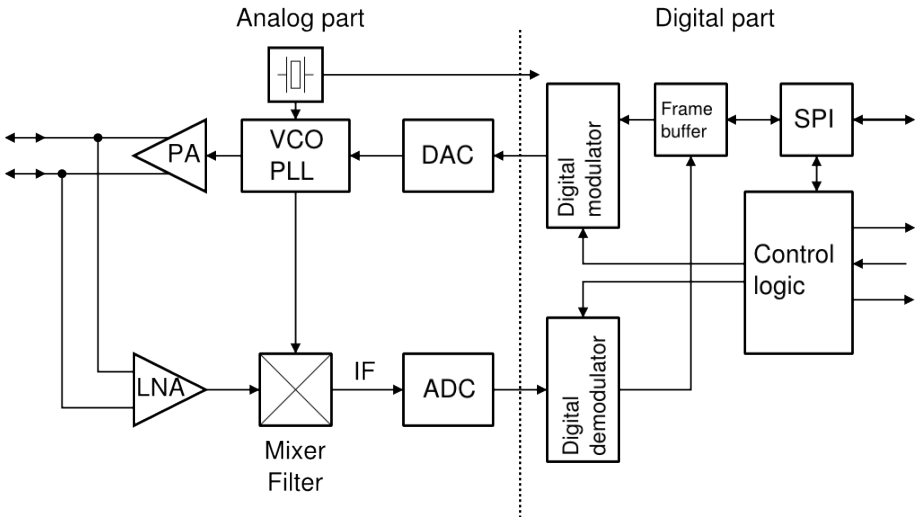


Abbildung 15: Vereinfachtes Blockschaltbild des AT86RF230

Der AT86RF230 wird durch eine Zustandsmaschine gesteuert. Abbildung 16 zeigt das Zustandsdiagramm für den Basic-Mode[30]. Zustandswechsel erfolgen nachdem Kommandos über die SPI-Schnittstelle geschickt wurden, in Abhängigkeit des SLP_TR-Pins oder durch Ereignisse in der Signalverarbeitung. Die Bedeutung der Zustände erklären die folgenden Abschnitte.

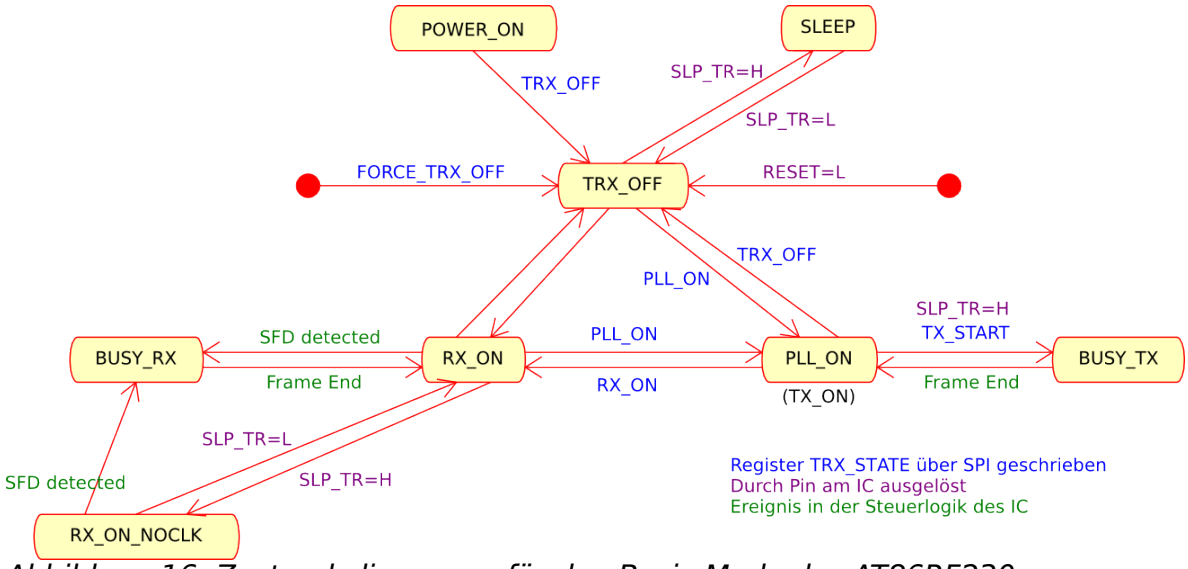


Abbildung 16: Zustandsdiagramm für den Basic-Mode des AT86RF230

Soweit das Grundlegende zum Chip. Im folgenden wird die Initialisierung des 2-Kanal-Senders erklärt. Im Datenblatt zum AT86RF230 sind im Abschnitt „Register Reference“ alle Funktionen der Register im Detail aufgelistet[31].

Initialisierung des 2-Kanal-Senders:

- I/O-Pins zur Kommunikation mit dem AT86RF230 initialisieren
- Register TRX_STATE mit dem Zustand TRX_OFF setzen (alle Register sind 8 Bit breit)
- In Register TRX_CTRL_0 setzt man den Prescaler für die Taktweiterleitung an den Mikrocontroller (Im Beispiel 4 MHz)
- Register PAN_ID_0 und PAN_ID_1 beinhalten die Pan-Id (16-Bit) anhand derer man eingehende Frames filtern lassen kann
- Register SHORT_ADDR_0 und SHORT_ADDR_1 enthalten die Short-Address (16-Bit) anhand derer noch strenger gefiltert wird
- Das Register PHY_TX_PWR beinhaltet unter anderem das Flag zum aktivieren der Auto-CRC-Funktion, womit dem MAC-Layer diese Aufgabe abgenommen wird
- Im Register PHY_CC_CCA legt man die Sendefrequenz fest, sowie auf welche Weise ein

Kanalfrei (CCA) ermittelt wird

- Im Register IRQ_MASK wird das Flag IRQ_TRX_END gesetzt, der Irq-Pin generiert nun einen Puls sobald eine Übertragung abgeschlossen ist oder ein kompletter Frame empfangen wurde
- Der TRX_STATE wird auf TX_ARET_ON gestellt. Der Zustand entspricht PLL_ON Basic-Mode, allerdings wird der vollständige CSMA/CA-Algorithmus durch den Chip übernommen wenn der SLP_TR-Pin eine Übertragung startet

Die Initialisierung des Empfängers ähnelt dem Sender. Der Empfänger unterscheidet sich lediglich im letzten Punkt. Stattdessen werden folgende Schritte durchgeführt:

- Der TRX_STATE wird auf PLL_ON gesetzt
- Nach einer Wartezeit bis zur Stabilisierung des PLL wird TRX_STATE schließlich auf RX_AACK_ON gesetzt. Der Zustand entspricht RX_ON im Basic-Mode, jedoch mit der Zusatzfunktion, dass bei Empfang eines Frames mit zum Gerät passender Adresse automatisch ein Acknowledge gesendet wird

Der Zugriff auf die die Register erfolgt über die SPI-Schnittstelle. Während einer Übertragung wird als erstes ein Command-Byte geschickt, gefolgt von einem einem Data-Byte das je nachdem gelesen oder geschrieben wird (Abbildung 17). Die Bitfolge auf dem Bus ist MSB zuerst.

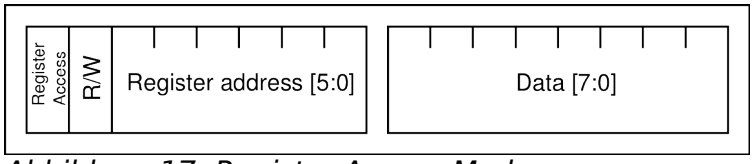


Abbildung 17: Register-Access-Mode

Das Hauptprogramm des Senders ist eine Endlosschleife. Eine Timer-Interruptroutine hilft beim Timing und schreibt eine Variable die das Hauptprogramm pollt. Daraus wird der Zeitpunkt zum Senden ermittelt. Der Status der Tasten wird abgefragt und an eine Funktion übergeben die das Byte in einen Frame verpackt, der dem MAC-Frameformat des IEEE-Standards entspricht. Das ist notwendig, denn im Extended-Mode prüft die Empfänger-Logik eingehende Daten und verwirft inkorrekte Frames.

Danach überträgt die Funktion den Frame in den Puffer des Chips und gibt das Signal zum Senden. Der IC wechselt darauf seinen Zustand von TX_ARET zu BUSY_TX_ARET solange er mit dem Übertragen der Daten beschäftigt ist. Abbildung 18 zeigt die Formatierung in der der Frame über die SPI verschickt wird. Das Frame-Length-Field

beinhaltet die Anzahl der Data-Bytes inklusive der zwei Bytes für die FCS (Frame-Check-Sequence = CRC) die der Chip automatisch generiert.

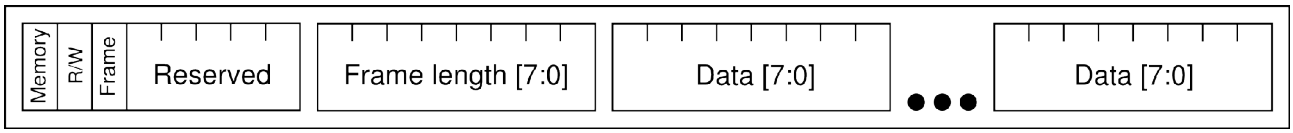


Abbildung 18: Tx-Frame-Mode

Auch beim Empfänger besteht das Hauptprogramm aus einer Endlosschleife und einer gepollten Timer-Variable. Der Empfänger erkennt eingehende Daten indem er das IRQ-Flag im AT86RF230 pollt. Ist das Flag gesetzt, wird der Frame aus dem Puffer gelesen und in den Mikrocontroller übertragen. Der Rx-Frame (Abbildung 19) unterscheidet sich vom Tx-Frame durch das LQI-Feld (Link-Quality-Indication), das nicht im Frame-Length-Feld mitgezählt wird.

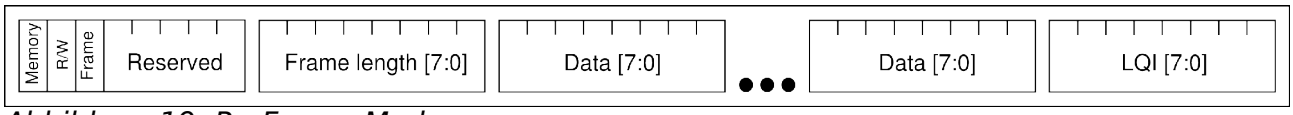


Abbildung 19: Rx-Frame-Mode

Abbildung 20 (Seite 33) zeigt Flussdiagramm mit dem CSMA/CA-Algorithmus, wie er im AT86RF230 nach dem Standard IEEE 802.15.4 realisiert ist. Kommt ein Kommando zum Senden, wechselt die Logik des ICs ihren Zustand, wartet eine zufällige Zeitspanne (Random-Exponetial-Backoff) und führt danach einen Kanalfrei-Test durch. Ist der Kanal frei, beginnt die Übertragung des Frames. Nach dem Ende der Übertragung startet ein Timer und der Transceiver schaltet auf Empfang um. Innerhalb eines kurzen Timeouts (~0,54 ms) muss ein gültiges Acknowledge-Pakte eingehen, dann war die Übertragung erfolgreich. Zwei Fehlerfälle können auftreten. Beim CHANNEL_ACCESS_FAILURE kam es zu keinem Kanalfrei-Ergebnis. Folglich ist der gewählte Kanal belegt und sollte gewechselt werden. Bei NO_ACK erfolgte keine Bestätigung auf gesendete Frames. Der gewählte PAN-Coordinator ist nicht mehr in Reichweite, ein neuer Coordinator sollte gesucht werden.

Der Empfang eines Pakets benötigt einen weniger Komplexen Algorithmus. Abbildung 21 (Seite 34) zeigt die Auto-Acknowledge-Funktion des AT86RF230. Im Zustand RX_AACK_ON wartet der Empfänger auf einen SOF (Start-Of-Frame). Detektiert der

Empfänger einen Frame wechselt er in den Zustand BUSY_RX_AACK_ON. Ist der Frame komplett im Puffer gespeichert findet eine Prüfung des MAC-Headers statt. Ist die Frame-Check-Sequence korrekt und die Adresse stimmt mit dem Register SHORT_ADDRESS_0/1 über ein, löst dies den TRX_END_IRQ aus. Weiter schaltet die Logik auf Senden um und überträgt ein Acknowledge-Paket. Der Transceiver wechselt wieder in den Zustand RX_AACK_ON und ist fertig.

Der CSMA/CA-Algorithmus hat eine bedeutende Auswirkung. Da der Random-Backoff vor jedem Senden erfolgt, hat das negativen Einfluss auf die Nettodatenrate. Der Wert für BE (Backoff-Exponent) liegt zwischen 3 und 5 und ergibt eine Wartezeit von 2 bis 10 ms (Gleichung 3). Hinzu kommt die Zeit zum Übertragen des Acknowledge, mit 192 µs (= 12 Symbole) für die Turn-Around-Time des Empfänger zum senden und 352 µs (= 11 Byte = 22 Symbole) für das Übertragen des Acknowledge-Paketes.

$$Backoff\ period = \left(2^{BE} - 1\right) \cdot Unit\ backoff\ period \cdot Symbol\ period$$
$$Max.\ backoff\ period = \left(2^{5} - 1\right) \cdot 20 \cdot 16\,\mu s = 9920\,\mu s$$
$$Min.\ backoff\ period = \left(2^{3} - 1\right) \cdot 20 \cdot 16\,\mu s = 2240\,\mu s$$

Gleichung 3: Formel zur Berechnung der minimalen und maximalen Backoff-Periode

Aufgrund des CSMA/CA-Algorithmus leidet die Netto-Übertragungsrate etwas sorgt aber für eine schnelle und zuverlässige Übertragung der Daten. Die Signalverarbeitung und das verwendete Modulationsverfahren sind sehr robust, damit ist der Datentransfer gut gegen Störeinflüsse abgesichert. Die Ansteuerung über SPI macht die Handhabung der Transceiver unproblematisch. Fertige Module ermöglichen den Aufbau einer Funkstrecke mit kleinstem Aufwand.

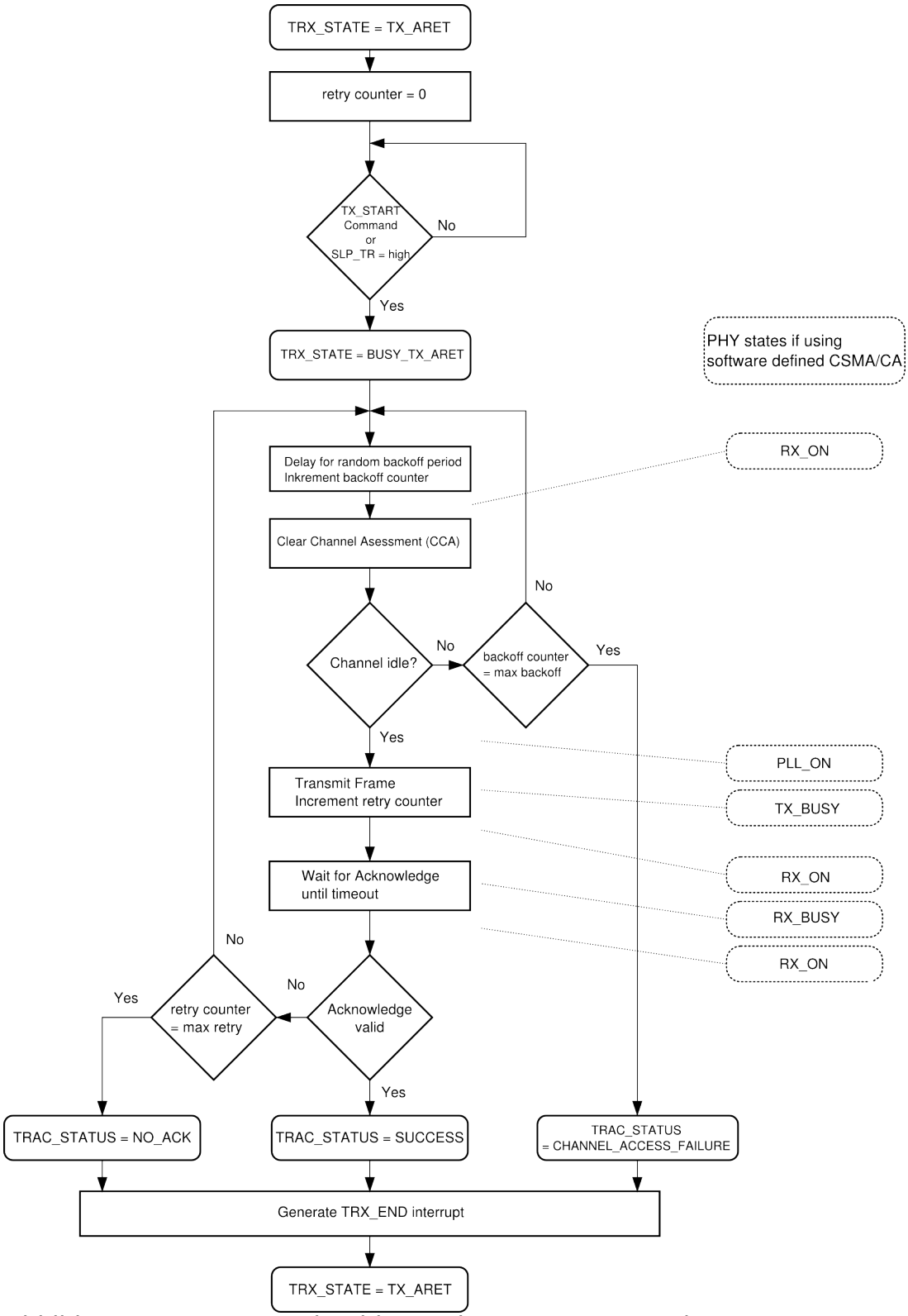


Abbildung 20: CSMA/CA-Algorithmus des AT86RF230 nach IEEE 802.15.4

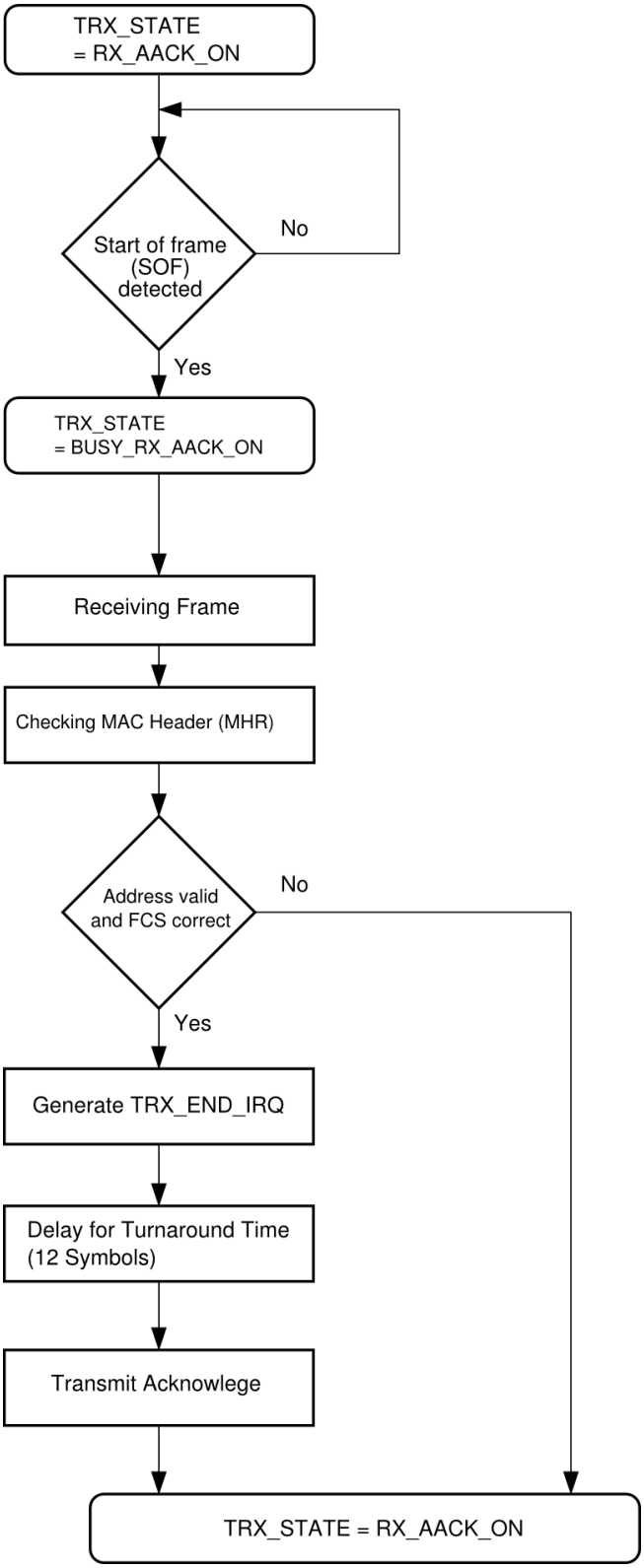


Abbildung 21: Auto-Acknowledge-Funktion des AT86RF230

7 Der MAC-Layer

Das nächste Abstraktionsniveau, auf der sich eine drahtlose Kommunikation umsetzen lässt, ist der MAC-Layer. Die durch den Standard IEEE 802.15.4 beschriebenen Eigenschaften bieten eine leistungsfähige Funktionalität auf Basis einer definierten Schnittstelle. Dagegen definiert der Standard nicht die konkrete Umsetzung und Verwendung des Stacks. Das hängt von der Implementierung ab. Software die für einen bestimmten Stack programmiert wurde lässt sich nicht ohne Aufwand auf einen anderen Stack übertragen. Eine allgemeine Beschreibung des MAC-Layers liefern die nächsten Abschnitte.

7.1 Data-SAP des MAC-Layer

Der Data-SAP (Service-Access-Point) des MAC-Layers dient zu Durchreichung von Daten. Zum SAP gehören die Primitives MCPS-Data-Request und MCPS-Data-Confirm (MCPS = MAC-Common-Part-Sublayer) zum Senden sowie MCPS-Data-Indication zum Empfangen von Daten. Hinzukommt MCPS-Purge-Request und MCPS-Purge-Confirm zum entfernen von Daten aus der Warteschlange im MAC-Layer[32].

Mit der Primitive *MCPS-Data-Request* überträgt man ein MSDU (MAC-Service-Data-Unit) an den MAC-Layer. Die Primitive enthält noch weitere Parameter. In den Feldern Address-Mode kann für Ziel- und Sender-Adresse zwischen drei Typen gewählt werden: keine Adresse, Short-Address (16-Bit, wird typischerweise vom PAN-Coordinator zugewiesen) oder Extended-Address (64-Bit, eine EUI-64 von IEEE verwaltete). Der Parameter Tx-Options ist ein Bit-Feld, in dem die angewandte Übertragungsmethode bestimmt wird. Dazu gehört das GTS-Flag (Guaranteed-Time-Slot) und bedeutet, das die Daten in einem zuvor zugewiesen Zeitschlitz übertragen werden, was allerdings von den meisten Stacks nicht unterstützt wird. Security-Enable weißt die verschlüsselte Übertragung an. Die weiteren Optionen zur Verschlüsselungs müssen vorher mit MLME-Set-Request im MAC-PIB gesetzt worden sein. Die Indirect-Transmission ist der bevorzugte Kommunikationsweg vom Coordinator zum Endgerät. Damit sind erst Stromsparmaßnahmen möglich, denn das Endgerät muss hier nicht ständig Empfangsbereit sein. Das Endgerät erfährt statt dessen über einen Eintrag im Beacon von verfügbaren Daten (Abbildung 22) oder frägt regelmäßig mit MLME-Poll-Request den Coordinator ab (Abbildung 23).

Das Acknowledge-Flag bestimmt ob die Übertragung ein ACK-Paket erfordert, denn genau genommen ist ein ACK nicht vorgeschrieben, sondern kann vom Sender gefordert werden. Der Parameter MSDU-Handle hat die Aufgabe, dass ein MSDU in der Warteschlange des MAC-Layers identifiziert werden kann.

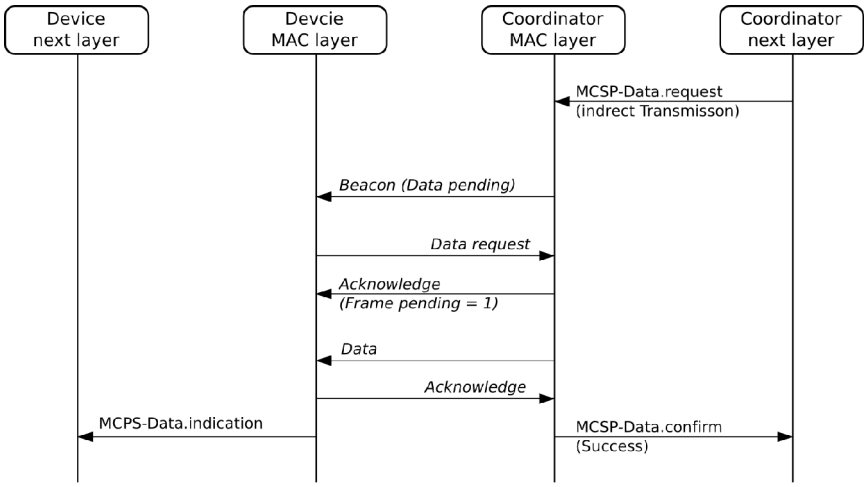


Abbildung 22: Indirect-Transmission mit Beacon

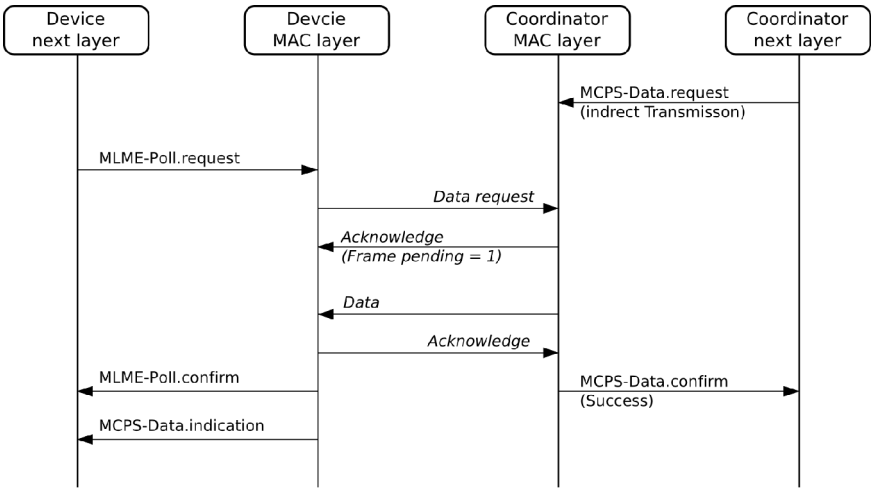


Abbildung 23: Indirect-Transmission ohne Beacon

Abbildung 24 auf der folgenden Seiten liefert einen Eindruck der notwendigen Schritte auf Ebene des PHY zum Übertragen von Daten.

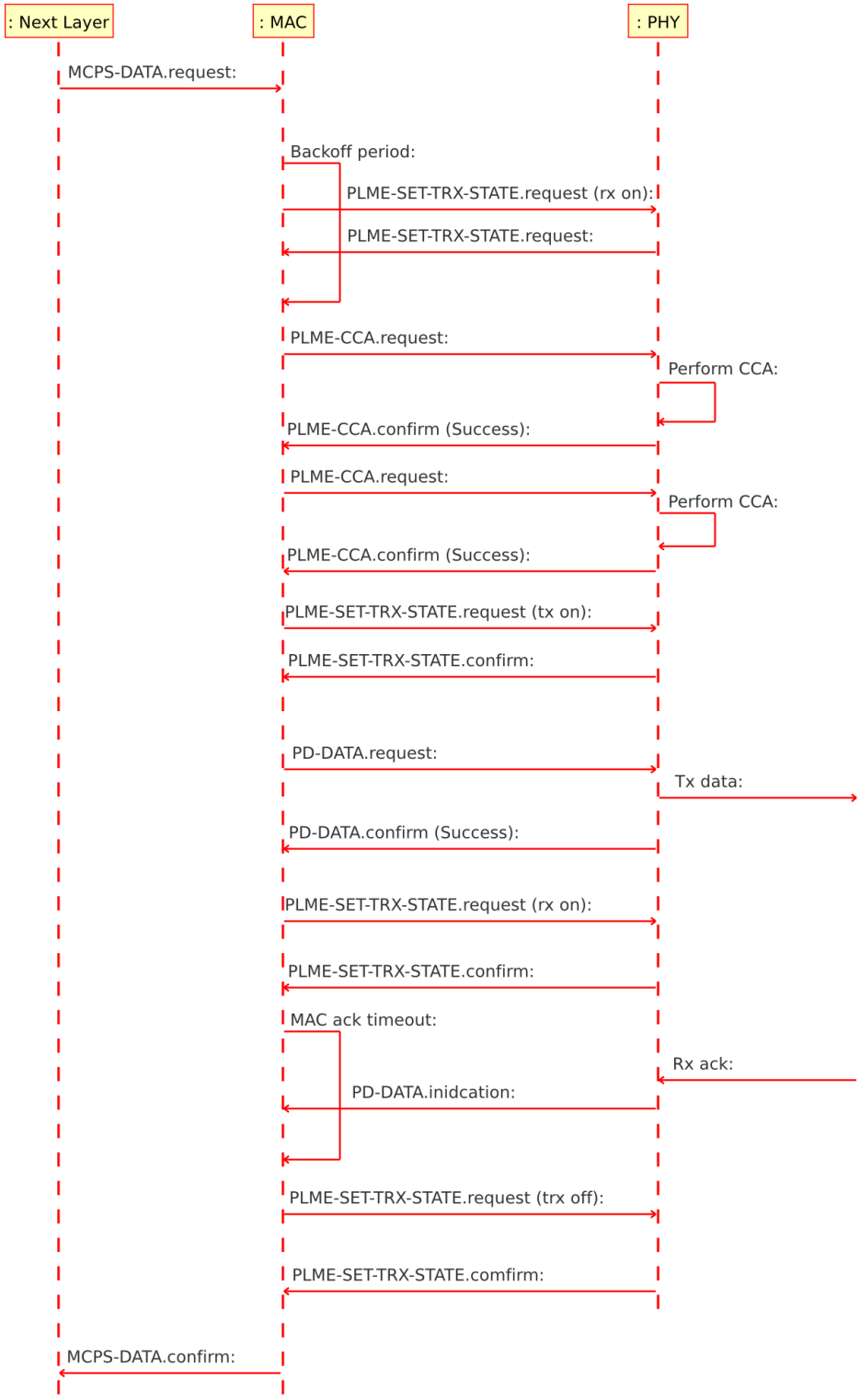


Abbildung 24: Abläufe nach einem MCPS-Data-Request

Empfangene Daten reicht der MAC-Layer über die *MCPS-Data-Indication*-Primitive zur nächst höheren Schicht weiter. Damit der Transceiver auf Empfangsbereitschaft schaltet,

muss der über einen MLME-Rx-Enable-Request dazu aufgefordert werden. Auf den Indication folgt kein Response, da keine Rückmeldung erforderlich ist. Das senden des Acknowledge-Pakets gehört in den Bereich des MAC-Layers.

Der *MCPS-Purge-Request* erlaubt es MSDUs, die im MAC-Layer noch in der Warteschlange stehen, anhand des Handles gezielt zu löschen. Diese Funktion müssen nur FFDs beherrschen und dient insbesondere zum Löschen von wartenden MSDUs mit der Option Indirect-Transmission, deren Ziel ein nicht mehr erreichbares Endgerät ist.

7.2 Management-SAP des MAC-Layer

Der Management-SAP des MAC-Layers liefert Dienste, die über das reine Übertragen von Daten hinaus gehen und für den Zugriff auf das Medium erforderlich sind. Mit dem absuchen aller Kanäle kann das richtige Netzwerk ausfindig gemacht werden. Durch Beacon-Frames synchronisiert der PAN-Coordinator Netzwerkteilnehmer zeitlich[33].

Mit dem *MLME-Associate-Request* (MLME = MAC-Layer-Management-Entity) fragt ein Endgerät den PAN-Coordinator um Erlaubnis im Netzwerk mit ihm teilzunehmen. *MLME-Associate-Indication* zeigt dem PAN-Coordinator an, dass ein Endgerät wünscht in sein Netzwerk einzusteigen. Über *MLME-Associate-Response* teilt dieser dem Endgerät mit welche Short-Address es im Netzwerk nun hat oder überhaupt teilnehmen darf. Die gesendete Adresse gelangt nun über *MLME-Associate-Confirm* zum Endgerät. MLME-Comm-Status-Indication informiert den PAN-Coordinator, ob die Übertragung der Antwort erfolgreich war. Abbildung 25 stellt die Abläufe nochmals grafisch dar.

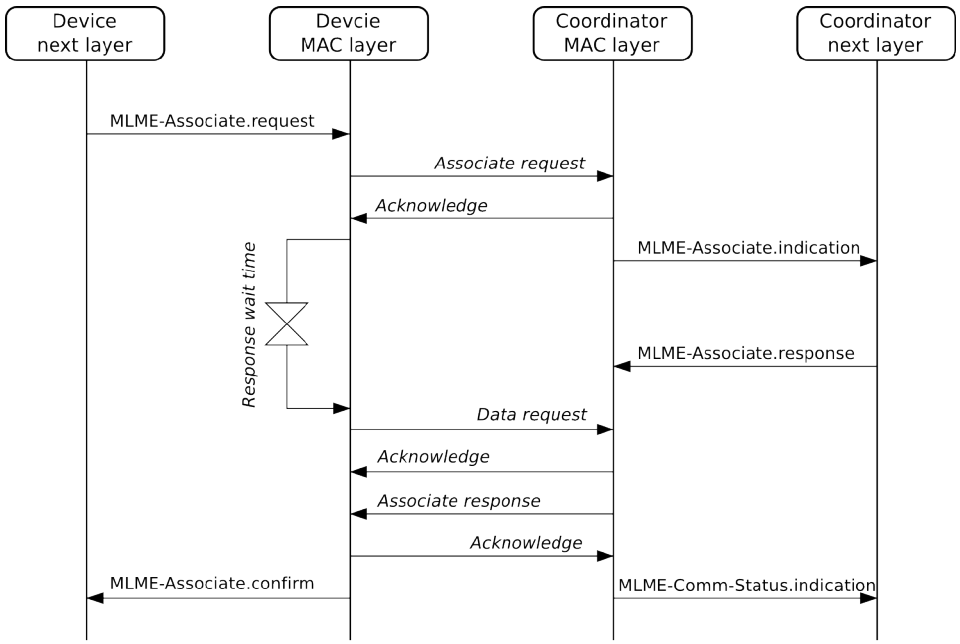


Abbildung 25: Abläufe nach einem MLME-Associate-Request

Mit dem *MLME-Disassociate-Request* meldet ein Endgerät, dass es das Netzwerk verlässt, umgekehrt kann auch der PAN-Coordinator zum verlassen des Netzwerks auffordern. Die Gegenseite erhält ein Indication über die Aufforderung. Ein Response ist nicht erforderlich, da die Bestätigung der Übermittlung durch ein ACK hier ausreicht.

Enthält ein eingegangener Beacon-Frame Daten löst das ein *MLME-Beacon-Notify-Indication* aus. Ist das Autorequest-Flag im MAC-PIB gesetzt wertet der MAC-Layer den Beacon-Frame selbstständig aus, ob Daten für Indirect-Transmission im PAN-Coordinator warten ansonsten führt auch das zu dem Indication.

Über den *MLME-Get-Request* lassen sich Informationen aus dem MAC-PIB abfragen. Der *MLME-Set-Request* setzt Werte im MAC-PIB. Der Abschnitt MAC-PIB (Seite 41) stellt einige der dort enthaltenen Variablen vor.

Benötigt ein Endgerät einen fest zugewiesenen Zeitschlitz in dem nur es senden darf, beantragt es diesen beim PAN-Coordinator mit *MLME-GTS-Request*.

Ein *MLME-Reset-Request* setzt alle Werte des MAC-PIB auf die Standard-Werte und der Transceiver geht auf den Zustand TRX_OFF.

MLME-Rx-Enable-Request weißt den MAC-Layer an den Transceiver auf Empfangsbereitschaft zu schalten. Der Parameter RxOnDuration legt die Dauer, fest für die der Receiver in Betrieb ist, angegeben in Symboldauer (16 µs bei 2,45-GHz).

Bevor irgendeine Kommunikation möglich ist, muss ein Kanal ausgewählt werden. Zur Gewinnung einer Entscheindungsgrundlage dient der *MLME-Scan-Request* und stellt mehrere Methoden zum Scannen zur Verfügung. Mit dem Energy-Detect-Scan kann der PAN-Coordinator einen freien Kanal ausfindig machen. Der Active-Scan überprüft durch Senden eines Beacon-Request, dass kein anderer PAN-Coordinator den Kanal benutzt. Endgeräte (RFDs) müssen die Fähigkeit des Active-Scan nicht besitzen, damit allerdings ein PAN ohne Beacon gefunden wird, ist die Funktion auch für Endgeräte sinnvoll. Beim Passive-Scan sucht ein Endgerät nach den Beacons, die ein PAN-Coordinator aussendet. Der Orphan-Scan hilft den Status eines Endgeräts aufzufrischen, bei dem das Limit an Übertragunsfehlern überschritten wurde.

Ein *MLME-Orphan-Indication* wird im PAN-Coordinator durch den Orphan-Scan eines Endgerätes, das zu ihm gehört, ausgelöst. Auf diese Weise kann ein Endgerät eventuell versäumte Änderungen der Netzwerkparameter erneut anfordern.

Der *MLME-Start-Request* teilt dem MAC-Layer mit, dass er die Rolle eines PAN-Coordinator übernimmt. Gleichzeitig legt der Request wichtige Netzwerkparameter fest, welcher Kanal wird genutzt, werden Beacons gesendet und wie lautet die PAN-Id.

Zur Kommunikation mit stromsparenden Geräten nutzt man die Indirect-Transmission. In Beacon-Enabled PANs teilt der Coordinator über die Beacon-Frames mit, für welche Geräte bei ihm Daten bereitliegen. Damit nun der MAC-Layer des Endgeräts weiß, dass es auf Beacon-Frames achten soll, instruiert man ihm das mit dem *MLME-Sync-Request*.

Sollte ein Endgerät für eine unzulässig langen Zeitraum keine Beacon-Frames empfangen, gibt das der MAC-Layer mit der *MLME-Sync-Loss-Indication* bekannt. Zu den weiteren Aufgaben der Schnittstelle gehört, zu melden, wenn der PAN-Coordinator die Parameter des Netzwerks geändert hat und erkennen das ein zweiter PAN-Coordinator mit der gleichen PAN-Id existiert.

Erfolgt die Abwicklung der Indirect-Transmission nicht über Beacon-Frames, lässt sich der

PAN-Coordinator mit einem MLME-Poll-Request abfragen. Sind Daten verfügbar werden diese automatisch abgeholt und ein MCPS-Data-Indication ausgelöst.

Als Übersicht stellt Tabelle 4 nochmal alle Primitives dar. Bei einigen Funktionen wurden nicht alle Primitives angesprochen, da zu einem Request im Allgemeinen ein Confirm gehört, aber nur als Bestätigung für das erfolgreiche Ausführen eines Requests dient.

Funktion	Request	Indication	Response	Confirm
MCPS-Data	x	x	x	x
MCPS-Purge	x			x
MLME-Associate	x	x	x	x
MLME-Disassociate	x	x		x
MLME-Beacon-Notify		x		
MLME-Get	x			x
MLME-GTS	x	x		x
MLME-Orphan		x	x	
MLME-Reset	x			x
MLME-Rx-Enable	x			x
MLME-Scan	x			x
MLME-Comm-Status		x		
MLME-Set	x			x
MLME-Start	x			x
MLME-Sync	x			
MLME-Sync-Loss		x		
MLME-Poll	x			x

Tabelle 4: Primitives des MAC-Layer

7.3 MAC-PIB

Das Verhalten des MAC-Layers bestimmen maßgeblich die Parameter im PIB (PAN-Information-Block) und bleiben nachdem ein Gerät mit einem PAN-Coordinator assoziiert ist im Wesentlichen unverändert. Der größte Teil der Einstellungen im PIB können auf ihrem Defaultwert belassen werden. Einige Beispiele sollen im folgenden den Zweck des PIBs verständlich machen und auf bedeutende Variablen hinweisen[34].

Die Parameter für den CSMA/CA-Algorithmus lassen sich im PIB festlegen. Der MaxCSMABackoffs-Wert schreibt vor, nach wie vielen Backoffdruchläufen der Kanal als belegt angesehen wird. MaxFrameRetries ist eine festgeschriebene Konstante und gibt

an wie oft ein Übertragungsversuch nach ausbleibendem Acknowledge wiederholt wird. Den Bereich in dem die Random-Backoffperiode liegt definieren MaxBE und MinBE (siehe auch Seite 32).

Damit der MAC-Layer nur Daten weiterreicht, die an ihn adressiert sind, speichert er die IEEE-Address und die durch den PAN-Coordinator zugewiesene Short-Address. Damit keine Überschneidung der Short-Address aus anderen PANs Probleme bereitet, ist auch die PAN-Id im PIB enthalten.

Dass Daten nach der Methode der Indirect-Transmission übertragen werden können wurde bereits ausgeführt. Noch nicht geklärt ist, was geschieht wenn ein Endgerät die Daten nicht abholt. Hierbei schafft die TransactionPersistenceTime abhilfe. Verbleibt ein Frame länger in der Warteschlange des MAC-Layers wird er verworfen. Die Default-Zeit liegt bei 7,68 Sekunden und der Maximalbetrag kann ca. 16 Minuten sein (siehe Formel in Gleichung 4)

$$BaseSuperframeDuration = BaseSlotDuration \cdot NumSuperframeSlots \cdot Symbol\ period$$
$$BaseSuperFrameDuration = 60 \cdot 16 \cdot 16\,\mu s = 15{,}36\,ms$$
$$PersistenceTime = TransactionPersistenceTime \cdot BaseSuperframeDuration$$
$$PersistenceTime(Default) = 500 \cdot 15{,}36\,ms = 7{,}68\,s$$
$$PersistenceTime(Max.) = 65535 \cdot 15{,}36\,ms \approx 1007\,s \approx 16\,min$$

Gleichung 4: Berechnungsgrundlage zur Persistence-Time

Zuletzt noch die schon häufig erwähnten Beacon-Frames. Die verschickt der MAC-Layer, nachdem er entsprechend Konfiguriert wurde, automatisch. Die Intervalle, in denen ein Beacon ausgesendet wird, hängt vom Wert in BeaconOrder (BO) ab. Die Formel in Gleichung 5 gibt den Einstellbereich des Beacon-Intervalls an. Wenn BO den Wert 15 einnimmt werden keine Beacons gesendet.

$$BeaconInterval = 2^{BO} \cdot BaseSuperframeDuration$$
$$BeaconInterval(Max.) = 2^{14} \cdot 15{,}36\,ms \approx 250\,s$$
$$BeaconInterval(Min.) = 2^{0} \cdot 15{,}36\,ms = 15{,}36\,ms$$

Gleichung 5: Berechnung des Beacon-Intervalls

Der MAC-Layer bietet umfangreiche Funktionen. Was zunächst übertrieben und penibel erscheint stellt doch die Grundlagen für eine flexible Funkschnittstelle, die viele sehr unterschiedliche Anforderungen gleichzeitig erfüllt. Zu denen gehören stromsparendes Arbeiten, aufgrund der Indirect-Transmission, wenn nötig schnelle Reaktionszeiten über direkte Übertragung, geordnetes aufspüren anderer Gerät usw.

8 Beispiel für die Anwendung des MAC-Layers

Damit die wichtigsten Elemente des MAC-Layers besser verstanden werden und das Zusammenspiel der einzelnen Funktionen klarer wird, vertieft das folgende Beispiel deren Anwendung.

Das „Drahtlose Abstimmgerät“ besteht aus einem Laboraufbau als Coordinator und zwei weiteren Laboraufbauten als Abstimmgerät. Die Software ist für bis zu 100 Teilnehmer ausgelegt. Nur Geräte deren IEEE-Adresse dem Coordinator bekannt sind können mit abstimmen. Eine Wahl läuft Zeitgesteuert ab, d. h. allen Abstimmgeräten wird die Dauer der Wahl mitgeteilt und beenden sie selbstständig. Die Ansteuerung erfolgt über die serielle Schnittstelle des PCs (siehe Abbildung 26).

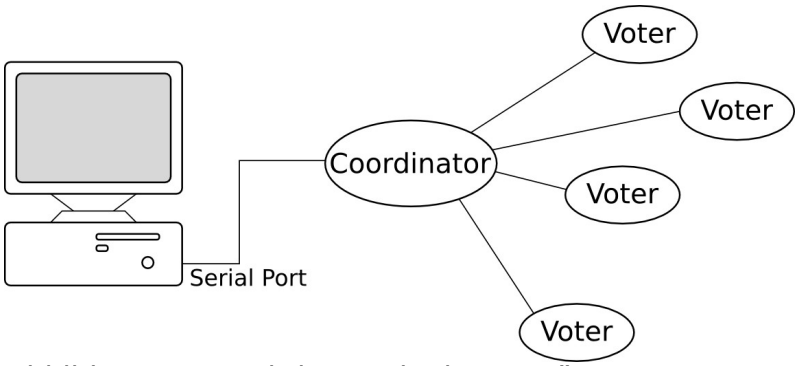


Abbildung 26: Drahtloses Abstimmgerät am PC

Anhand dieser Anwendung lassen sich die Phasen zeigen mit denen der MAC-Layer gestartet wird. Es wird der Ablauf verdeutlicht der nötig ist um ein Gerät zu assoziieren und disassoziieren. Des weiteren werden die zur Datenübertagung notwendigen Schritte abgehandelt.

8.1 Entwicklungsumgebung

Die Hardware des Laboraufbaus wurde bereits vorgestellt (Kapitel 5 Seite 23). Nun fehlen noch Angaben zur Software. Der verwendete C-Compiler ist der avr-gcc (Version 4.1.2). Als Standard-Bibliothek kommt avr-libc (Version 1.4.4) zum Einsatz. Für ihre Module stellt Meshnetics einen kostenlosen 802.15.4-Stack zur Verfügung. Download des Stacks „OpenMAC“ (Version 1.4.1.1) ist über die Webseite von Meshnetics oder bei

SourceForge.net[35] möglich. Die angebotene Software liefert ein komplettes Framework inklusive HAL. Alle genannten Programme stehen unter verschiedenen Opensource-Lizenzen deren Bedingungen große Freiheit bieten.

Den Aufbau der OpenMAC-Plattform zeigt Abbildung 27. Meshnetics setzt als Betriebssystem TinyOS ein. Diese fertige Umgebung ist leicht zu handhaben. TinyOS ist ein ereignisgesteuertes Betriebssystem, als Hauptprogramm dient ein periodischer Timer-Event und ähnelt damit der Endlosschleife in einfacheren Programmen. Aufrufe für Request- und Response-Primitives an den MAC-Layer erfolgt über Funktionsaufrufe, die ein Ereignis in der Warteschlange setzen und sofort zurückkehren. Indication und Confirm-Primitives sind durch in der OpenMAC-Bibliothek nicht definierte Funktionsprototypen realisierte Callback-Funktionen. In der Anwendung müssen entsprechend benannte Funktionen deklariert werden. Beim Build-Prozess zu einem ausführbaren Programm löst der Linker die Deklarationen zu einem gewöhnlichen Funktionsaufruf auf.

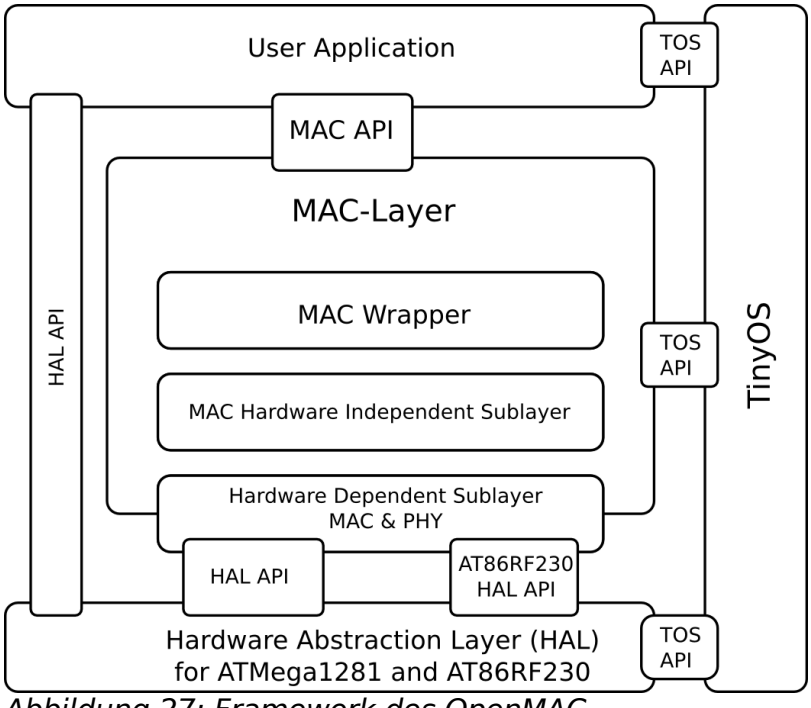


Abbildung 27: Framework des OpenMAC

Welche Daten das heruntergeladene Tarball-Paket, das den OpenMAC, enthält weißt die in Tabelle 5 aufgeführte Verzeichnis-Struktur aus. Der MAC-Layer und TinyOS sind in NesC geschrieben. Für beide sind vorkompilierte Bibliotheken enthalten, die sich über

separate Header-Dateien mit dem C-Compiler avr-gcc nutzen lassen. Die Beispielprogramme und der HAL sind für Evaluation-Board von Meshnetics und Atmel ausgelegt. Dank eines gut geschriebenen HAL kann der auch für den Laboraufbau direkt genutzt werden.

./src/HAL/HAL_R5/base/ ./src/HAL/HAL_R5/eZeeNet/	HAL für ZigBit-Modul mit AVR und AT86RF230 (in C)
./src/HAL/HAL_R5/meshBean2/ ./src/HAL/HAL_RZ200_RCB/rcb/ ./src/HAL/HAL_STK500/stk500/	HAL für externe Komponenten auf dem Evaluation-Board MeshBean 2 von Meshnetics sowie RZ200 und STK500 von Atmel (in C)
./src/Stack/development/PHY/PHY_R3/IEEE802.15.4/	Von Hardware abhängiger Teil des MAC- und PHY-Layer (in NesC)
./src/Stack/development/MAC/MAC_R3/IEEE802.15.4/	MAC-Layer (in NesC) und MAC-C-Wrapper
./src/TOSLib/	Bibliothek für TinyOS (in NesC)
./src/SampleApplication/	Beispielanwendungen in C und NesC
./userdoc/readme.htm	Kurze Beschreibung des OpenMAC

Tabelle 5: Verzeichnisstruktur des OpenMAC

Aus den gegebenen Projekten lassen sich eigene ableiten, indem ein neues Verzeichnis in Ordner ./src/SampelApplication/ angelegt wird und die Dateien eines Beispiel-Projekt dorthin kopiert. Weiter sind Anpassungen an der Makefile notwendig. Die Zeile `COMMON_LIBS += -lZigBitExt` legt fest das die HAL-Bibliothek für das ZigBit-Modul mit externer Taktquelle (Anzapfung des AT86RF230) genutzt wird. Alternativ stehen HAL-Bibliotheken für das ZigBit-Modul mit internem RC-Oszillator (ZigBitInt) und für die Atmel-Hardware RZ200 und STK500 zur Auswahl.

Nach Vorgabe des OpenMAC ist für die Nutzung mit externer Taktquelle die Low-Fuse des AVRs auf 0xc0 und die High-Fuse auf 0x9e zu stellen, bei internem RC-Oszillator sollten die Fuses auf 0xc2 und 0x9e stehen.

Die MAC-Bibliothek steht in drei Version bereit, MACLibF (Full Function Device, FFD), MACLibR (Reduced Function Device, RFD) und MACLibS (Spezialversion für Diagnosezwecke). Ein einfaches das „Hello-World“-Programm das mit dieser Software compiliert und in den Controller programmiert wird erreicht bereits eine Codegröße von 35 kB. Eine beträchtliche Größe, bei intensiver Nutzung der avr-libc kommen ca. 10 kB hinzu.

Im Detail lässt sich ein Standard aus verschiedenen Gründen oft nicht umsetzen. Ähnlich anderen MAC-Bibliotheken unterstützt OpenMAC keine Guaranteed-Timeslots[36] und die Auswertung des sog. Frame-Pending-Bits im Acknowledge-Paket findet nicht statt. Dass MLME-Sync fehlt, fällt schwerer ins Gewicht. Somit kann eine Indirect-Transmission nur über MLME-Poll abgeholt werden.

8.2 Erstellen eines eigenen Projekts

Die Projekt-Dateien für das Beispiel „Drahtloses Abstimmgerät“ basieren auf der mitgelieferten Beispielanwendung „LowPower“ und wurden von dort in das Unterverzeichnis „voter“ kopiert. Einige Anpassungen an der Makefile mussten noch vorgenommen werden. Sowohl Coordinator als auch Endgerät arbeiten mit der ZigBitExt-Bibliothek. Die Targets wurden in „coord“ und „voter“ umbenannt. Einige C-Konstanten wurden definiert die für das Projekt erforderlich sind und das Build-Verzeichnis wurde umbenannt. Die Erstellung der Programm erfolgt mit den Aufruf `make mik_board`. Der Download der Software erfolgt seperat mit einem beliebigen ISP-Programmiergerät.

8.3 Starten des MAC-Layers

Nun zur Funktion des drahtlosen Abstimmgeräts. Nachdem der Startup-Code von TinyOS durchlaufen wurde erfolgt der einmalige Aufruf der Callback-Funktion `void appEntry__main(void)`. Das ist der richtige Platz zur Initialisierung der Hardware. Die serielle Schnittstelle wird geöffnet und für 38,4 kBaud konfiguriert, darf aber nicht sofort genutzt werden, denn der Port-Enable löst ein Start-Bit auf der Leitung aus. Ein Timer wird in TinyOS geöffnet und eine Callback-Funktion als Function-Pointer übergeben. Diese stellt später das Hauptprogramm dar. Der Timer wird dabei noch nicht gestartet. Als letztes erfolgt der Aufruf `MAC_init()` eine OpenMAC spezifische Funktion deren Ausführung ein Ereignis in die Warteschlange von TinyOS setzt. Nachdem das Ereignis abgearbeitet wurde folgt der Aufruf der Callback-Funktion `MAC_initDone()` durch TinyOS.

Abbildung 28 und 29 zeigen die Etappen in denen der Coordinator startet und auf den MAC-Layer zugreift. Die erste Etappe initialisiert den MAC-Layer und dessen PIB. Mit der zweiten Etappe erfolgt das Scannen der verfügbaren Kanäle. Im Beispiel wählt ED-Scan

den Kanal aus der die geringste Signalstärke aufweist. Der Active-Scan stellt sicher, dass kein anderes Gerät mit der selben PAN-Id in dem gewählten Kanal vorkommt. Andernfalls wird erneut ein ED-Scan durchgeführt und ein anderer Kanal ausgewählt. Bei Erfolg startet der MLME-Start-Request den MAC-Layer als Coordinator und schaltet auf Empfangsbereitschaft. Der Timer für das Hauptprogramm wird gestartet.

Abbildung 28: Initialisierung

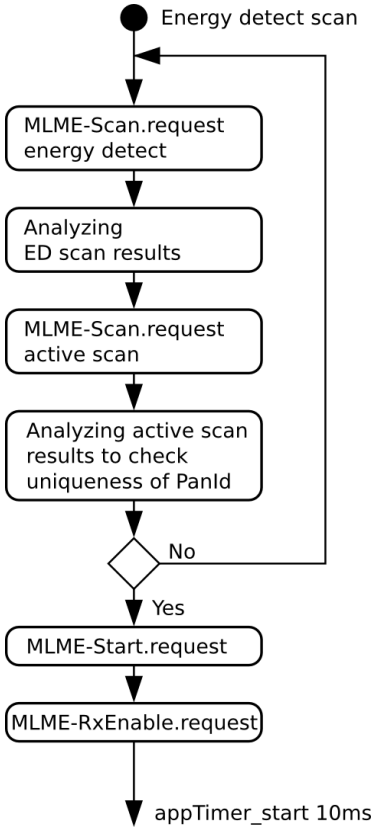


Abbildung 29: Scan

8.4 Funktionsweise des Coordinator

Nachdem der Coordinator hochgefahren ist, befindet er sich im Bereitschafts-Zustand. Von hier ab können Kommandos über die serielle Schnittstelle an das Gerät geschickt werden. Abbildung 30 zeigt unter welchen Umständen der Coordinator seinen Zustand

wechselt. Je nach Zustand ändert sich das Verhalten gegenüber den Votern.

Im Zustand START_NETWORK können sich die Voter über einen Associate-Request beim Coordinator registrieren. Sind alle Geräte aus der Teilnehmertabelle im Eeprom anwesend, ist das Netzwerk vollständig und ein Übergang in den Zustand COORD_IN_NETWORK erfolgt.

Nun kann mit dem Kommando „>votestart %i“ eine Abstimmung gestartet werden. Wobei „%i“ die Zeit in Sekunden angibt, die die Wahl dauern soll (Übergang zu VOTING). Ist die Abstimmung abgeschlossen muss abgewartet werden bis die Stimmen von allen Endgeräten eingesammelt ist (VOTE_CLOSED). Liegen alle Stimmen vor, dann geht der Coordinator in den Zustand VOTE_RESULT. Auf das Kommando „>voteresult“ wird das Wahlergebnis ausgegeben und gleichzeitig findet die Transition zu COORD_IN_NETWORK statt. Tabelle 5 enthält eine Kurzreferenz der wichtigsten Befehle für die Steuerung des Coordinator.

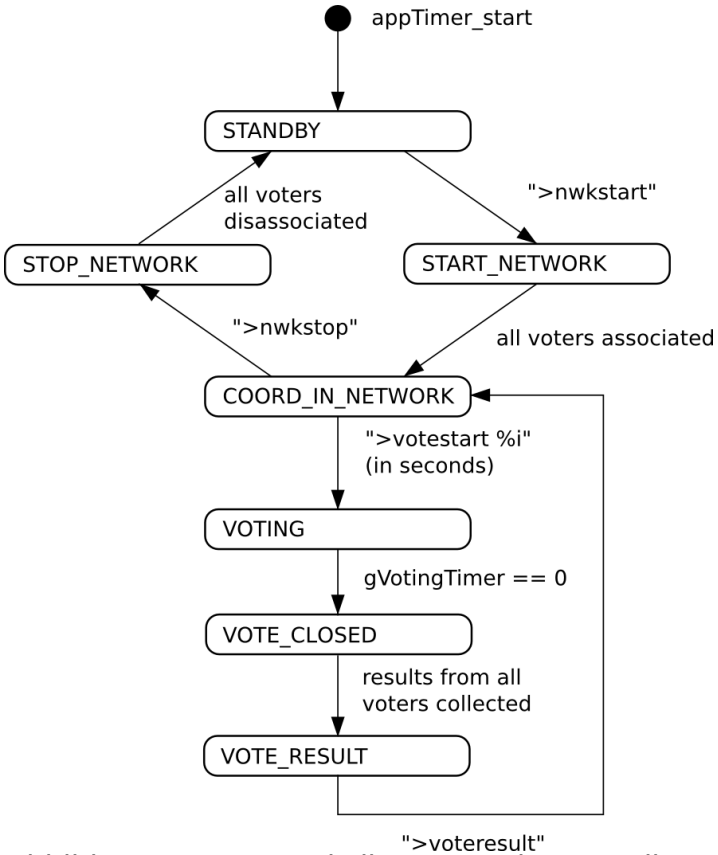


Abbildung 30: Zustandsdiagramm des Coordinator

Der Coordinator führt nicht nur für sich eine Zustandsmaschine, sondern merkt sich auch den Status der mit ihm verbundenen Endgeräte (Abbildung 31). Ein erfolgreich assoziiertes Gerät befindet sich im Zustand `IN_NETWORK`, wenn beim Abfragen mehrfach kein Acknowledge zurück kam, wird sein Zustand in DEVICE_MISSING geändert. Wurde kein Verbindungsversuch unternommen verbleibt das Gerät als `NOT_INITIALIZED` markiert.

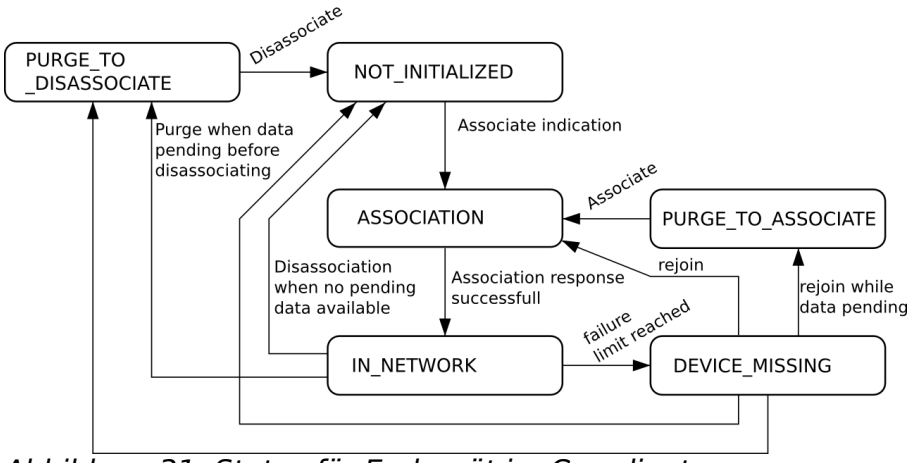


Abbildung 31: Status für Endgerät im Coordinator

>nwkstart	Association der Voter zulassen
>nwkstop	Verbindung mit allen Votern trennen
>votestart %i	Start eines Wahlgangs, %i Dauer der Wahl in Sekunden
>voteresult	Ausgabe des Ergebnis nach Ablauf der Wahl
>setenddev %i %a3 %a2 %a1 %a0	Voter in die Adressliste aufnehmen, %i Eintrag in der Adressliste, %a3-0 IEEE-Adresse des Voters, Hexadezimal in Vierergruppen

Tabelle 6: Kurzreferenz für Coordinator

8.5 Associate

Die Aufnahme eines Teilnehmer kann nicht ohne Prüfung seines Status vorgenommen werden. Damit ein Voter im Netzwerk mitmachen darf, fragt er mittels des MLME-Associate-Request den Coordinator um Erlaubnis. Für das weitere sorgt der Coordinator. Abbildung 32 zeigt den Ablauf im Coordinator anhand eines Flussdiagramms. Das beschreibt nicht den Ablauf einer einzelnen Funktion, sondern einen Vorgang über mehrere Callback-Funktionen hinweg verläuft.

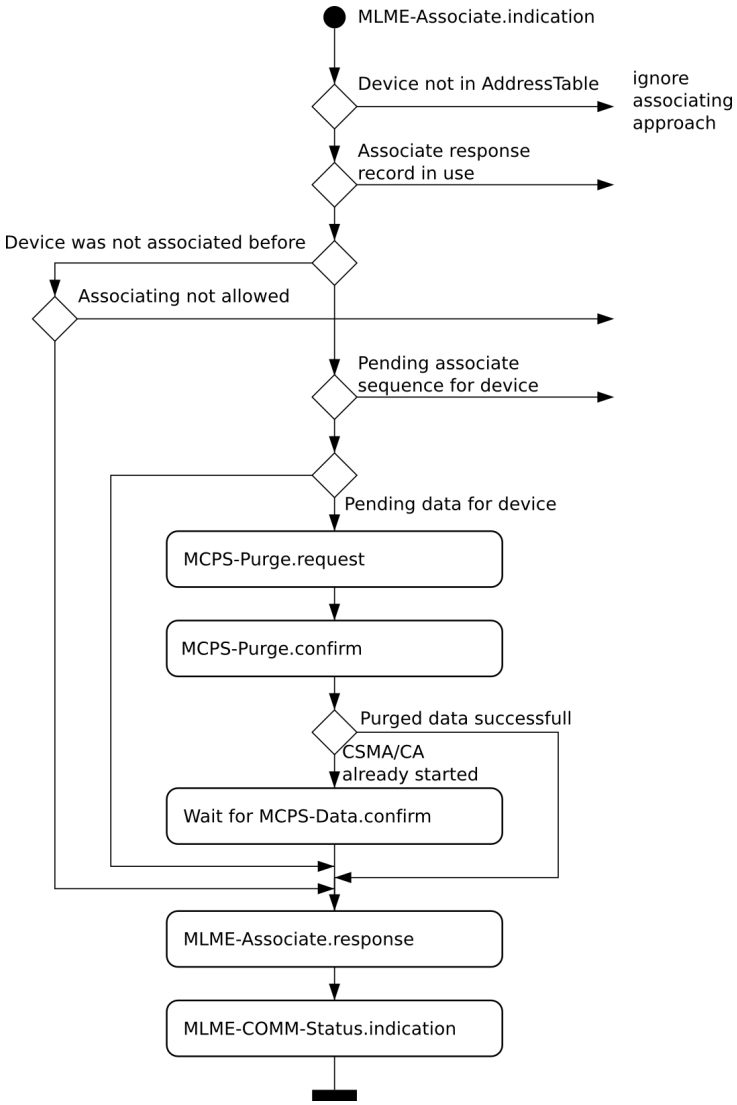


Abbildung 32: Flussdiagramm für Association

Zunächst wird abgefragt, ob das Gerät in der Liste der erwünschten Teilnehmer steht. Nicht berechtigte Geräte werden ignoriert. Nach einem Timeout wird der Anmeldeversuch durch das anfragende Gerät als erfolglos abgebrochen. Gemäß Standard kann eine Gerät auch unter Angabe eines Grundes (z. B. PAN voll oder Access-Denied[37]) abweisen werden. Ein Mikrocontroller verfügt über nur wenig Speicher, daher sind die vorhandenen Ressourcen stark begrenzt und müssen frei sein, ansonsten wird die Registrierung abgebrochen. Sofern Neuanmeldungen erwünscht sind beantwortet der Coordinator die Anfrage schließlich mit einem MLME-Associate-Response. MLME-Comm-Status-Indication gibt den Status zurück, mit dem die Übertragung der Antwortnachricht abschloss. Außerdem kann es vorkommen, dass ein Gerät versucht sich anzumelden, obwohl es bereits angemeldet war. Wenn der Anmeldeversuch noch

läuft bleibt die Anfrage unberücksichtigt. Ansonsten wird geprüft, ob gerade Daten in der Warteschlange stehen und ggf. entfernt und der Teilnehmer erneut registriert. Ist der CSMA/CA-Algorithmus bereits im Gange kann ein Sendeversuch nicht mehr gestoppt werden, es muss bis zu dessen Abschluss gewartet werden. Hiernach wird der Status des Geräts initialisiert und mit einem Response, entsprechend einem normalen Associate-Versuch, wieder aufgenommen.

8.6 Datenübertragung

So bald ein Teilnehmer registriert ist, schickt der Coordinator regelmäßig Nachrichten, auf die eine prompte Antwort zu erfolgen hat (Abbildung 33). Bleibt ein Acknowledge oder eine Antwort mehrfach aus, setzt der Coordinator für das Gerät den Status `DEVICE_MISSING` und pollt es nicht mehr bis es seine Anwesenheit mit einer Associate-Anfrage wieder klarstellt.

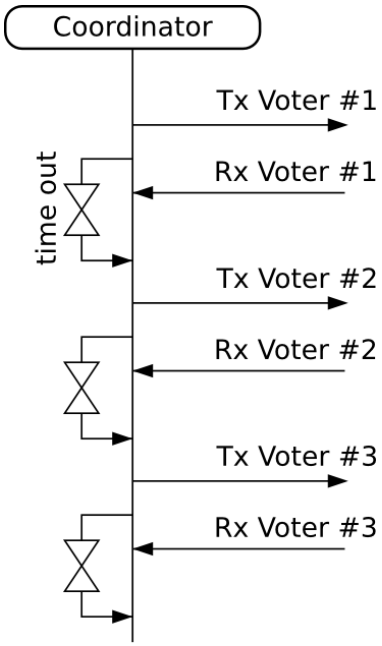


Abbildung 33: Pollsequenz

Soweit zur prinzipiellen Funktion. Anhand der Datenübertragung soll das auslösen einer Primitive im OpenMAC gezeigt werden. Listing 1 zeigt eine Funktion, die an `SendData()` aus dem Quellcode des Coordinators angelehnt ist. Zuerst wird die Datenstruktur `gDataReq` mit den Parametern der Primitive ausgefüllt. Die OpenMAC spezifische

Funktion `MSRV_getAffixLength()` berechnet die Länge des MAC-Headers. Der Footer besteht aus der Prüfsumme, die der Transceiver-Chip berechnet und wird deswegen hier nicht berücksichtigt. Danach werden die zu versendenden Daten in das MSDU kopiert. Zusätzlich muss vor dem MSDU freier Speicher mit der Länge des MAC-Headers reserviert sein. Das erspart dem MAC-Layer zeitraubendes Umkopieren der Inhalte beim Anfügen des Headers.

```
/*************************************************************************
  Sends data to the enddevice.
  params: data - pointer to the data buffer, length - data length.
  returns: result of MCPS_DATArequest.
*************************************************************************/
result_t SendData(uint8_t* pData, uint8_t length)
{
  MACFrameParams frameParams;

  gPollSequenceState = PS_DATA_SENDING_STATE; // Statemachine

  gDataReq.srcAddrMode = MAC_SHORT_ADDR_MODE;
  gDataReq.srcPANId = gCoord.PanId;
  gDataReq.srcAddr.sh = gCoord.ShortAddress;
  gDataReq.dstAddrMode = MAC_SHORT_ADDR_MODE;
  gDataReq.dstPANId = gCoord.PanId;
  gDataReq.dstAddr.sh = gCurrentDeviceShortAddress;
  gDataReq.msduLength = length;
  gTxHandle += 1;
  gDataReq.msduHandle = gTxHandle;
  gDataReq.txOptions = MAC_ACK_TXOPTION;
  frameParams.type = MAC_DATA_TYPE;
  MSRV_getAffixLength(&frameParams);
  gDataReq.msdu = gpTransactionBuffer + frameParams.header;
  memcpy(gDataReq.msdu, pData, length);
  return MCPS_DATArequest(&gDataReq);
}
```

Listing 1: Request auslösen

Viele Primitives die an den MAC-Layer geschickt werden kopieren die mittels Pointer übergebenen Daten nicht, z. B. MCSP-Data-Request oder MLME-Associate-Request. Das bedeutet es müssen Daten übergeben werden, deren Lebensdauer über die Ausführung der Funktion `SendData()` hinaus gehen. Erst nach Rückmeldung, etwa durch eine Confirm-Primitive, wird der Speicher nicht mehr durch den MAC-Layer beansprucht. Durch diese Maßnahme beansprucht der Stack nur wenig des knappen Speichers und die Speicherverwaltung verbleibt Zentral bei der darüberliegenden Software.

Vom MAC-Layer aufgerufene Callback-Funktionen erhalten die Daten auch als Pointer. Ihre Gültigkeit beschränkt sich aber auf den Scope der Funktion (Listing 2). Der Vorteil nur den Pointer, statt der gesamten Daten, als Parameter weiter zu geben, liegt im höheren Rechenaufwand. Denn dann müssen die Daten nicht in den Programm-Stack umkopiert werden.

```
/*************************************************************************
  Notifies that data frame has been recieved.
  indParams - params of received data frame.
  IEEE802.15.4 paragraph: 7.1.1.3
*************************************************************************/
void MCPS_DATAindication(MACDataIndParams* params)
{
  // code
}
```

Listing 2: Callback-Funktion einer Indication-Primitive

8.7 Funktion des Voters

Die Initialisierung des MAC-Layers als Endgerät unterscheidet sich gegenüber dem Coordinator. Das Flussdiagramm in Abbildung 34 und 35 stellt den Ablauf dar. Die Initialisierungsphase ist kurz und besteht im wesentlichem aus einem Reset-Request. Zur Suche des Coordinator wird ein Active-Scan über alle Kanäle durchgeführt. Stimmt die PAN-Id mit der gesuchten über ein, wird ein Associate-Request ausgeführt, ansonsten startet ein neuer Suchdurchlauf. Gleiches geschieht wenn der Associate-Request fehlschlägt. Bei Erfolg muss auf Empfangsbereitschaft geschaltet werden, damit Daten vom Coordinator erhalten werden können.

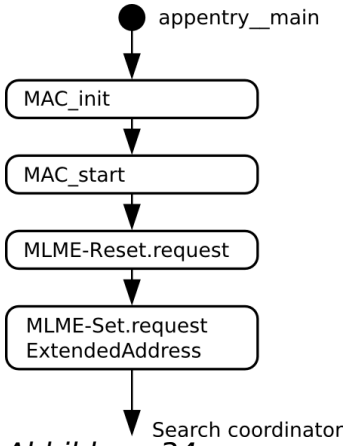


Abbildung 34: Initialisierung

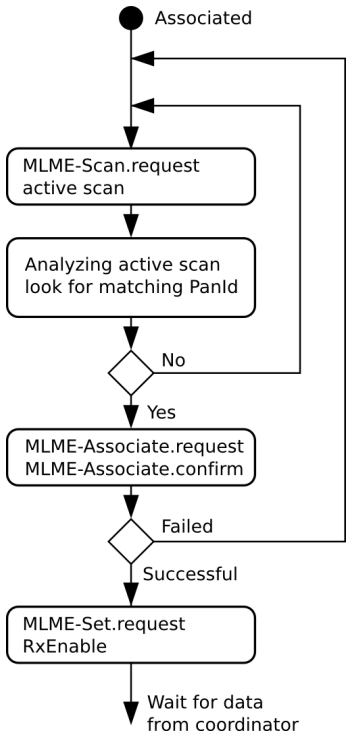


Abbildung 35: Scan

Nach dem die Initialisierung und Anmeldung abgeschlossen ist, erhält der Voter vom Coordinator regelmäßig Daten. Auf diese Weise erfährt das Gerät wann ein Wahlgang stattfindet. Eine Abstimmung dauert immer eine vorher festgelegte Zeit. Solange die Abstimmung läuft befindet sich der Voter im Zustand `VOTE_RUNNING` (Abbildung 36), bzw. sobald eine Stimme abgegeben wurde in `VOTE_CASTED` und eine LED gibt dem Wähler die Bestätigung seiner Eingabe. Nach Abgabe der Stimme kann sie noch bis zum Ablauf der Wahl geändert werden. Nach dem Ende der Wahl verbleibt der Voter im Zustand `VOTE_CLOSED`, bis der Coordinator alle Stimmen eingesammelt hat. Anschließend wechselt der Status wieder in die Ausgangsposition `IN_NETWORK`.

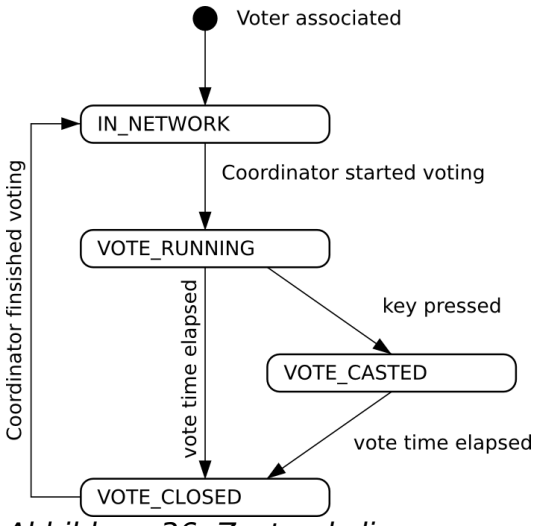


Abbildung 36: Zustandsdiagramm des Voters

In diesem Kapitel wurden einige Aspekte der Anwendung des MAC-Layers gezeigt, die besonderer Erklärung bedurften. So z. B. das ein Purgen bei laufendem CSMA/CA-Algorithmus nicht mehr möglich. Bedeutend ist auch die Erkenntnis, dass für den MAC-Layer aufgrund dessen Komplexität, die Umsetzung ein einfaches Betriebssystem als Rahmen benötigt. Die Umsetzung des OpenMAC ist insgesamt gelungen und eignet sich sehr gut als Grundlage für Funkanwendungen.

9 Ausblick

ZigBee und 802.15.4 sind eine weites Feld und bieten noch viel Raum für Untersuchungen. An erster Stelle sollte die Frage der Koexistenz mit WLAN und Bluetooth geklärt werden bzw. Grenzsituationen, bis zu denen ein Parallelbetrieb möglich ist, ermittelt werden. Ab welcher Teilnehmerdichte können Störungen durch WLAN auftreten und kann der Fall auch in der Praxis auftreten? Kann industrielles Bluetooth (20 dBm Sendeleistung) oder eine sehr nahe beim Empfänger aufgestellte Freisprecheinrichtung die Übertragung beeinträchtigen?

Im Bezug auf ZigBee lässt sich noch vieles erforschen. Der nächste Schritt wäre eine einfache Anwendung auf Grundlage eines seriell angesteuerten ZigBee-Moduls. Im speziellen wären hierfür die ZigBit-Module mit der Software SerialNet geeignet (wird nur zusammen mit einem Evaluation-Kit ausgeliefert[38]), sowie XBee Series 2[39] von Maxstream geeignet. Die Ansteuerung der Module könnte hierbei über den PC erfolgen. Als Anwendungsbeispiele wären die Funktion Lampe-Lichtschalter oder Übertragung von Messdaten denkbar.

Im Internetauftritt der ZigBee Alliance zeigt sich Bewegung. So stellt die Seite „Designed for ZigBee“[40] Produkte vor, die in naher Zukunft dem Standard entsprechen. Zusammen mit „Success Stories“[41] demonstriert dies breite des Anwendungsgebiets. Die überschaubare Liste zeigt aber auch, wie wenige die Umsetzung geschafft haben und damit welches Niveau die Implementation von ZigBee erfordert. Denn im Vergleich dazu umfasst die Mitgliederliste der ZigBee Alliance mehrere hundert Firmen.

Die kürzlich verabschiedete Spezifikation „Home Automation Public Application Profile“ schafft weiter Anreize für ZigBee. Sie ebnet den Weg für nahtlose Zusammenarbeit zwischen den Endgeräten und nicht nur die gemeinsamen Nutzung eines Netzwerks.

10 Literatur

IEEE 802.15.4-2003	http://standards.ieee.org/getieee802/download/802.15.4-2003.pdf
IEEE 802.15.4-2006	http://standards.ieee.org/getieee802/802.15.html
Zigbee Specification	http://zigbee.org/en/spec_download/zigbee_downloads.asp
ZigBee Application Profiles	http://zigbee.org/en/spec_download/zigbee_downloads.asp
Hands-On ZigBee	Fred Eady. Butterworth Heinemann 2007. ISBN-13: 978-0123708878
Datenblatt Atmega1281V	http://atmel.com/dyn/resources/prod_documents/doc2549.pdf 2549K–AVR–01/07
Datenblatt AT86RF230AVR	http://atmel.com/dyn/resources/prod_documents/doc5131.pdf 5131C-ZIGB-05/22/07
Datenblatt ZigBit-Modul	http://meshnetics.com/ZigBit_OEM_Module_Product_Datasheet.pdf DOC.M-251~01 v.1.9

10.1 Einzelnachweise

1 IEEE 802.15.4-2003 - Kapitel E.3, Analysen

2 IEEE 802.15.4-2003 - Anhang F

3 Werbung auf der Z-Wave Webseite http://www.z-wavealliance.org/modules/start/

4 Funkschau 11/2006, „Interferenz im 2,4-GHz-Band“ http://www.funkschau.de/heftarchiv/pdf/2006/fs_0611/fs_0611_s12.pdf

5 zigbee.org, „ZigBee and Wireless Radio Frequency Coexistence“ http://zigbee.org/en/markets/whitepapers1.asp

6 IEEE 802.15.4-2003 - Kapitel E.3.1.1

7 Datenblatt zu CC2420 von Texas Instruments (SWRS041), http://www.ti.com/lit/gpn/cc2420

8 Datenblatt Fractus Compact Reach Xtend (UM_FR05-S1-N-0-102), http://www.fractus.com/sales_documents/FR05-S1-N-0-102/UM_FR05_S1_N_0_102.pdf

9 Atmel Referenz Microstrip-Antenne für AT86RF230, http://www.atmel.com/dyn/resources/prod_documents/doc8095.pdf

10 Datenblatt „ZigBit OEM-Modules ZDM-1281-*“ von Meshnetics (DOC.M-251~01 v.1.9)

11 Texas Instruments Referenzantenne für CC2420 (AN040), http://www.ti.com/litv/pdf/swra093a

12 eazix, http://eazix.com/pro_zigbee_module.htm

13 Ember, http://www.ember.com/products_index.html

14 Jennic, http://www.jennic.com/products/index.php

15 MaxStream, http://www.maxstream.net/products/

16 MeshNetics, http://meshnetics.com/

17 Radiopulse, http://www.radiopulse.co.kr/

18 Renesas, http://eu.renesas.com/fmwk.jsp?cnt=zigbee_landing.jsp&fp=/products/connectivity/zigbee/

19 Sentec, http://www.sentec-elektronik.de/

20 Telegesis, http://www.telegesis.com/

21 ZigBee.org http://zigbee.org/en/certification/

22 ZigBee.org, „Download ZigBee Technical Documents“ http://zigbee.org/en/spec_download/zigbee_downloads.asp

23 Artikel „MAC address“ in der englischen Wikipedia http://en.wikipedia.org/wiki/MAC_address

24 http://standards.ieee.org/regauth/faqs.html#q15

25 Datenblatt für ATmega1281V

26 Datenblatt „ZigBit OEM-Modules ZDM-1281-*“

27 Datenblatt „ZigBit OEM-Modules ZDM-1281-*“, Seite 16

28 IEEE 8015.4-2003, Kapitel 6.5 „2450 MHz PHY specifications“

29 Wikipedia, Artikel „Direct Sequence Spread Spectrum“ und „Quadraturphasenumtastung“

30 Datenblatt für AT86RF230, Figure 7-1 , Seite 18

31 Datenblatt für AT86RF230, „Register Reference“, Seite 59

32 IEEE 802.15.4-2003 - Kapitel 7.1.1, MAC data service

33 IEEE 802.15.4-2003 - Kapitel 7.1.2, MAC management service

34 IEEE 802.15-4-2003 - Kapitel 7.4, MAC constants and PIB attributes

35 SOURCEFORGE.NET, http://sourceforge.net/projects/openmac/

36 Texas Instruments, 802.15.4 MAC API, F8W-2005-1503, http://focus.ti.com/docs/toolsw/folders/print/timac.html

37 IEEE 802.15.4-2003 - Kapitel 7.3.1.2.3, Association status field

38 Meshnetics ZigBit Development Kit, http://meshnetics.com/dev-tools/zdk/

39 Maxstream XBee Series 2, http://www.digi.com/products/wireless/zigbee-mesh/xbee-series2-module.jsp

40 Designed for ZigBee, http://zigbee.org/en/certification/designed_for_zigbee.asp

41 Success Stories, http://zigbee.org/en/markets/success_stories.asp

11 Abkürzungsverzeichnis

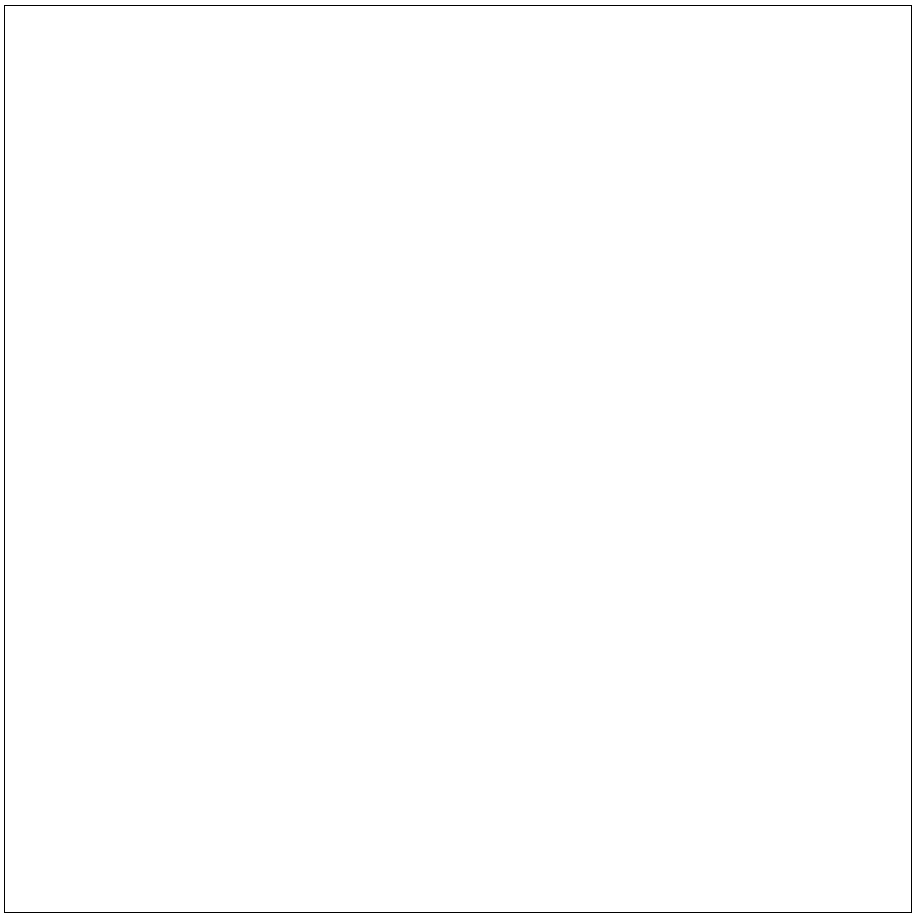
ACK	Acknowledge
AFH	Adaptive Frequenzy-Hopping
API	Application-Interface
APS	Application-Layer-Support
AVR	Advanced Virtual RISC
CCA	Clear-Channel-Assesment
CSMA/CA	Carrier-Sense-Multiple-Access - Collusion-Avoidance
BE	Backoff-Exponent
BI	Beacon-Intervall
BO	Beacon-Order
dBi	Antennengewinn im Bezug zum idealen Isotropstrahler
dBm	Pegelangabe mit Bezugsleistung 1 mW
EUI	Extended-Unique-Identifier
FCS	Frame-Correction-Sequenz
FFD	Full-Function-Device
FR4	Leiterplattenmaterial: Epoxidharz + Glasfasergewebe
GTS	Guaranteed Timeslot
HAL	Hardware-Abstraction-Layer
IAB	Individual-Address-Block
IB	Information-Block
IEEE	Institute of Electrical and Electronics Engineers
ISM	Industrial-Scientific-Medical
LQI	Link-Quality-Indication
MAC	Medium-Access-Control
MCPS	MAC-Layer-Common-Part-Sublayer
MLME	MAC-Layer-Management-Entity
MPDU	MAC-PDU
MSDU	MAC-SDU
NWK	Network
OUI	Organisation-Unit-Identifier
PAN	Personal-Area-Network
PDU	Protocol-Data-Unit
PER	Package-Error-Rate
PIB	PAN-Information-Block
PHY	Physical
PPDU	Physical-PDU
PSDU	Physical-SDU
RFD	Reduced-Function-Device
SAP	Service-Access-Point
SDU	Service-Data-Unit
SoC	System on Chip
SOF	Start-of-Frame
UWB	Ultra-Wide-Band
ZDO	ZigBee-Device-Object

12 Index

Stichwortverzeichnis

Active-Scan.....40
Adapterplatine.....27
Antenne.....9
Application-Object.....19
Application-Support-Sublayer.....19
Associate.....38, 50
avr-gcc.....44
avr-libc.....44
Balun.....9
Beacon-Intervall.....42
Callback-Funktion.....45
Chipantenne.....10
CSMA/CA-Algorithmus. .31
Disassociate.....39
Eigenentwicklung.....16
Energy-Detect-Scan.....40
EUI-64.....16
Full-Function-Device.....19
Fuse.....46
HAL.....45
HV-Programmierung.....25
IEEE-64-Bit-Adresse.....16
Indirect-Transmission.....35
Information-Block.....21
ISM-Band.....6
Laboraufbau.....23
MAC-Layer.....19, 35
MAC-PIB.....41
Makefile.....46
MCPS.....35
Microstrip.....10
MLME.....38
Module.....13
NesC.....45
Nettodatenrate.....32
Network-Layer.....19
Nutzungsstufen.....14
OpenMAC.....44
Orphan-Scan.....40
PAN-Coordinator.....19
Passive-Scan.....40
Physical-Layer.....20
Poll-Request.....36
Primitive.....21
Protocol-Data-Unit.....22
Protokollstapel.....18
Random-Backoff.....32
Reduced-Function-Device 19
Request.....21
Response.....21
SDK.....13
Service-Access-Point.....20
Service-Data-Unit.....22
Sleep-Mode.....24
Stromverbrauch.....26
Sync-Request.....40
TinyOS.....45
Transaction-Persistence-Time.....42
Transceiver-Chips.....11
WLAN.....6
Zertifizierung.....16
ZigBee-Device-Object...19
ZigBit.....24

13 Daten-CD



./Adapterplatine/	Layout-Daten für Adapterplatine, erstellt mit Eagle-CAD
./Bilder/	Im Rahmen der Diplomarbeit erstellte Bilder
./Datenblätter/	Datenblätter für Hardware und ZigBit-Modul
./Diplomarbeit/	Diplomarbeit in ODT- und PDF-Format
./Software/	Beispielanwendungen
./Software/2-Kanal/	Beispiel zur Ansteuerung der Hardware
./Software/OpenMAC/	Beispiel zur Nutzung des OpenMAC
./Standard/	IEEE 802.15.4-2003 und ZigBee-2004

14 Erklärung

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Augsburg, den 17. Dezember 2007

Michael Brenner